

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизації проектування енергетичних процесів та систем

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль

«___» _____ 2019 р.

Магістерська дисертація

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Система розпізнавання жестів для людино-машинної взаємодії

Виконала: студентка II курсу, групи ТВ-81мпз

Конкіна Наталія Сергіївна

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник: к.т.н. Шалденко Олексій Вікторович

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Консультант _____
(назва розділу) (вчені ступінь та звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ - 2019

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Освітньо-кваліфікаційний рівень «магістр»

Напрямок підготовки 121 „Інженерія програмного забезпечення”

Спеціальність «Програмне забезпечення розподілених систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль

(підпис)

” ____ ” _____ 2019 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

_____ Конкіній Наталії Сергіївні

(прізвище, ім'я, по батькові)

1. Тема дисертації _____ “Система розпізнавання жестів для людинно-машинної взаємодії”

Науковий керівник дисертації _____ Шалденко Олексій Вікторович, к.т.н., ст.викл.

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від 01.11.2019р. № 3812-с

2. Строк подання студентом роботи: 09.12.2019.

3. Об'єкт дослідження: методи, алгоритми і програми захоплення, відстеження та розпізнавання жестів людини.

4. Предмет дослідження: типи жестів, структура методів і алгоритмів захоплення, відстеження та розпізнавання динамічних жестів, їх взаємозв'язок, складність,

надійність, стійкість, що дозволяють розпізнавати динамічні жести в реальному часі.

5.Перелік завдань які потрібно розробити: здійснити порівняльний аналітичний огляд існуючих методів захоплення, відстеження та розпізнавання динамічних жестів людини, провести класифікацію жестів виконуються людиною і вибрати алфавіт динамічних жестів, придатний для створення людино-машинного інтерфейсу для управління комп'ютером, розробити обчислювально ефективний алгоритм захоплення і відстеження руки людини, розробити обчислювально ефективну модель і алгоритм розпізнавання динамічних жестів людини, провести експерименти по оцінці надійності і працездатності системи в реальному часі, що підтверджують теоретичні результати.

6. Перелік ілюстраційного матеріалу: існуючі системи розпізнавання жестів, архітектура можелі нейронної мережі, горизонтальне масштабування, UML діаграма класів, UML діаграма прецендентів.

7. Дата видачі завдання: " 9 " вересня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	12.09.2018 – 10.10.2018	
2.	Вивчення та аналіз задачі	10.10.2018 – 10.01.2019	
3.	Розробка архітектури та загальної структури системи	10.01.2019 – 25.01.2019	
4.	Розробка структур окремих підсистем	25.01.2019 – 20.02.2019	
5.	Підготовка матеріалів	20.02.2019 – 01.03.2019	
6.	Програмна реалізація системи	01.03.2019 – 31.05.2019	
7.	Оформлення пояснювальної записки	01.09.2019 – 22.11.2019	
8.	Захист програмного продукту	25.10.2019	
9.	Передзахист	22.11.2019	
10.	Захист	19.12.2019	

Студент Конкіна Н.С.
(прізвище та ініціали,)

(підпис)

Науковий керівник роботи Шалденко О.В.
(прізвище та ініціали,)

(підпис)

РЕФЕРАТ

Структура та обсяг дипломної роботи. Магістерська дисертація складається зі вступу, шести розділів, висновку, переліку посилань з 37 найменувань, 1 додаток, і містить 23 рисунків, 23 таблиць. Повний обсяг магістерської дисертації складає 92 сторінки.

Актуальність теми. Актуальність теми дисертації з теоретичної точки зору диктується необхідністю розробки методів, моделей і алгоритмів захоплення, відстеження та розпізнавання жестів, що здійснюються людиною в реальному часі, зокрема руками, придатних для створення інтерфейсу управління роботою комп'ютера з їх допомогою.

Мета роботи і завдання дослідження. Розробка загальної методології захоплення, відстеження та розпізнавання динамічних жестів людини, що здійснюються руками, включаючи моделі, методи і алгоритми, теоретичне і експериментальне обґрунтування працездатності цієї методології в реальному часі з високим рівнем надійності для створення працездатних людино-машинних інтерфейсів.

Для реалізації цієї мети були поставлені такі завдання:

1. Здійснити порівняльний аналітичний огляд існуючих методів захоплення, відстеження та розпізнавання динамічних жестів людини.
2. Провести класифікацію жестів виконуються людиною і вибрати алфавіт динамічних жестів, придатний для створення людино-машинного інтерфейсу для управління комп'ютером.
3. Розробити обчислювально ефективний алгоритм захоплення і відстеження кисті людини
4. Розробити обчислювально ефективну модель і алгоритм розпізнавання динамічних жестів людини.
5. Провести експерименти по оцінці надійності і працездатності системи в реальному часі, що підтверджують теоретичні результати.

Об'єкт дослідження. Методи, алгоритми і програми захоплення, відстеження та розпізнавання жестів людини.

Предмет дослідження. Типи жестів, структура методів і алгоритмів захоплення, відстеження та розпізнавання динамічних жестів, їх взаємозв'язок, складність, надійність, стійкість, що дозволяють розпізнавати динамічні жести в реальному часі.

Наукова новизна одержаних результатів. Було розроблено алгоритм розпізнавання жестів у відеопотоці з використанням нейронних мереж, розроблена система, яка надає необхідний інструментарій для розпізнавання жестів, досліджено особливості побудови моделі з використанням теорії розв'язання винахідницьких задач.

Практичне значення одержаних результатів. Сформульовані основні концепції проектування та реалізації алгоритму розпізнавання жестів з використанням методів та підходів побудови нейронних мереж, досліджено теорію розв'язання винахідницьких задач.

Ключові слова. *Розпізнавання жестів, нейронна мережа, теорія розв'язання винахідницьких задач, pyTorch, python, каскади Хаара.*

ABSTRACT

Structure and scope. The master's thesis consists of an introduction, six sections, a conclusion, a list of links of 37 titles, 1 appendix, and contains 37 figures, 23 tables. The full volume of the master's thesis is 92 pages.

Relevance. The relevance of the thesis topic from the theoretical point of view is dictated by the need to develop methods, models and algorithms of capture, tracking and recognition of hands gestures carried out by a person in real time, in particular by hands, suitable for creating an interface to manage the work of the computer with their help.

Purpose of the work and objectives of the research. Development of a general methodology for capturing, tracking and recognition of human dynamic gestures performed by hands, including neural networking models, methods and algorithms, theoretical and experimental justification of the performance of this methodology in real time to create workable human-machine interfaces. Studying existing methods of capture, tracking and recognition of human dynamic gestures.

To achieve this goal, the following tasks were set:

1. Develop a comparative analytical review of existing methods of capture, tracking and recognition of human dynamic gestures.
2. Develop classification of gestures of the person and choose the alphabet of dynamic gestures, suitable for creation of the human-machine interface for computer control.
3. Develop a computationally effective algorithm for capturing and tracking the human hand.
4. Develop a computationally effective model and algorithm for recognition of human dynamic gestures.
5. Perform experiments to assess the reliability and performance of the system in real time, confirming the theoretical results.

Object of research: methods, algorithms and programs of capture, tracking and recognition of human gestures.

Research subject: types of gestures, structure of methods and algorithms of capture, tracking and recognition of dynamic gestures, their interrelation, complexity, reliability, stability, recognition of dynamic gestures in real time.

Scientific novelty of the obtained results. The algorithm of recognition of gestures in a video stream with the help of neural networks has been developed, the system which gives necessary toolkit for recognition of gestures is developed, features of construction of model with use of the TRIZ investigated. The basic concepts of designing and implementing the algorithm of gesture recognition using methods and approaches of building neural networks are formulated, the theory of solving inventive problems is studied.

Practical value of the research. The basic concepts of design and implementation of gesture recognition algorithm using methods and approaches of neural networks construction are formulated, the TRIZ theory is investigated.

Key words. *Gesture recognition, neural network, inventive problem solving theory, pyTorch, python, Haar cascades.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1. МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ РУК ДЛЯ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ	12
2. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ РУК	15
2.1 Опис сенсору LeapMotion	15
2.2 Опис браслету Muo	16
2.3 Опис сенсору Kinect	17
2.4 Розпізнавання рухів на основі аналізу різниць зображень (MEI)	18
2.5 Розпізнавання конфігурації на основі аналізу гістограм напрямків	19
2.6 Розпізнавання конфігурації і позиції із застосуванням кольорових рукавичок.....	19
2.7 Розпізнавання конфігурації і позиції на основі аналізу контуру зображення руки	20
2.8 Розпізнавання жестів руки із застосуванням штучних нейронних мереж.....	22
2.9 Аналіз існуючих методів розпізнавання жестів руки на основі аналізу тривимірної моделі руки.....	23
Висновки до розділу 2	24
3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ	25
3.1 Особливості побудови моделі для нейронної мережі з використанням теорії розв’язання винахідницьких задач	25
3.2 Опис інструментів розробки.....	36
3.3 Обґрунтування вибору програмної реалізації	40
4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	44
4.1 Опис функціональності системи	44
4.2 Розробка методів розпізнавання жестів рук в системах людино-машинної взаємодії.....	46

4.3 Розробка алгоритму розпізнавання жестів на основі нейромережевої технології.....	47
4.5 Створення інсталяційного файлу.....	51
5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ ...	54
5.1 Інсталяція та системні вимоги	54
5.2 Інструкція з використання програмного продукту	57
6. СТАРТАП ПРОЕКТ	61
6.1 Основні ідеї проекту.....	61
6.2 Технологічний аудит ідеї проекту	63
6.3 Аналіз ринкових можливостей запуску стартап-проекту	64
6.4 Розроблення ринкової стратегії проекту	70
6.5 Розроблення маркетингової програми проекту	73
Висновки.....	76
Список використаних джерел.....	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД	—	база даних
СІ	—	безпосередня інтеграція
CD	—	безпосередня доставка
VPC	—	віртуальна приватна мережа
ПЗ	—	програмне забезпечення
ТРВЗ	—	теорія рішення винахідницьких задач
Плагін	—	розширення функціональності
ШНМ	—	штучна нейронна мережа

ВСТУП

Інформаційні технології - невід'ємна частина нашого життя. Внаслідок швидкого розвитку методів комунікації між людьми, неможливо залишити осторонь проблему комунікації із зовнішнім світом. Розпізнавання жестів дозволить людині спілкуватися і взаємодіяти з пристроями і машинами природно, без будь-яких додаткових пристроїв. Розпізнавання жестів дозволить управляти різними інженерними пристроями, може буде використано у медицині або у сфері розваг.

У зв'язку з удосконаленням обчислювальних потужностей і появою великих баз зображень стало можливим навчати глибокі нейронні мережі, які завдяки багат шаровій архітектурі дозволяють обробляти і аналізувати великий обсяг даних, а також моделювати когнітивні процеси в різних областях. У задачі розпізнавання образів особливого успіху досягли згорткові нейронні мережі (Convolutional Neural Networks). Завдяки нейронним мережам вдається досягти кращих результатів в більшості складних завдань, які класичними методами не вирішуються. Нейромережі адаптивні, завдяки чому для вирішення різних задач можна використовувати один інструмент.

Альтернативою згортковим мережам для розпізнавання об'єктів на зображенні є метод ознак (каскадів) Хаара. Інакше він ще називається методом Віюлі-Джонса за іменами дослідників, які адаптували метод вейвдетів Хаара до розпізнавання образів.

Поява сенсора Kinect в 2010 році, відкрило широкі можливості для створення систем розпізнавання жестів, що надзвичайно підвищило актуальність завдання розпізнавання жестів з використанням камери глибини. А розроблене для Kinect програмне забезпечення визначає положень основних суглобів тіла людини [1].

Для знаходження контуру руки з використанням бібліотек комп'ютерного зору, використовується алгоритм оконтурювання зображення, запропонований Satoshi Suzuki [2], після чого з усіх отриманих контурів вибирається той, в якому знаходиться точка долоні. Алгоритм на цьому кроці, отримуючи на вхід контур руки

людини, апроксимує його опуклим багатокутником. Для цього використовується механізм, запропонований Jack Sklansky [3].

Метою роботи є дослідження і розробка методів розпізнавання статичних і динамічних жестів руки, які можуть бути використані для безконтактної взаємодії людини з комп'ютером. Засобом досягнення мети є вирішення таких основних завдань:

- Аналіз існуючих методів відстеження та розпізнавання жестів руки, а також безконтактного людино-машинного взаємодії;
- дослідження і розробка методів опису, вилучення і розпізнавання конфігурацій руки;
- розробка нових методів розпізнавання жестів руки на основі нейронних мереж для управління комп'ютерними системами.

Базуючись на проведених дослідженнях, був розроблений власний програмно-апаратний комплекс для розпізнавання жестів рук та збереження результатів у базу даних з метою подальшої обробки та інтерпретації. Був розроблений власний аудіоплеєр, що керується за допомогою жестів.

1. МЕТОДИ РОЗПІЗНАВАННЯ ЖЕСТІВ РУК ДЛЯ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ

Людино-комп'ютерна взаємодія (англ. Human-computer interaction, HCI) - полідисциплінарний науковий напрям, що існує і розвивається з метою вдосконалення методів розробки, оцінки та впровадження інтерактивних комп'ютерних систем, призначених для використання людиною, а також з метою дослідження різних аспектів цього використання[5].

Одним із методів HCI, який отримав широке поширення в останні роки, є взаємодія, заснована на жестах людини [6, 7, 8].

Жести свідчать про інтенсивність переживань, про якість і спрямованість відносин, про культурну та групову приналежність. Жести, як і міміка, можуть нести самостійну інформацію про людину незалежно від її мови. Жести супроводжують мову або замінюють її; при цьому вони говорять про ставлення людини до якогось особи, події або предмету.

У HCI жести використовуються для передачі інформації в комп'ютер, який в подальшому може використовуватися для ідентифікації людини, управління комп'ютером, літальним апаратом, ігровим аватаром і т.д.

Розпізнавання жестів можна застосовувати в таких областях діяльності людини, як наприклад:

1. Управління комп'ютером і побутовими приладами

а. У комп'ютерному додатку кожної конфігурації руки зіставляється певна команда. Людина показує жест, система розпізнає конфігурацію руки і відправляє відповідну команду в комп'ютер.

б. Позиція долоні зіставляється з позицією курсору миші на екрані. Рухи руки призводять до рухів курсору. Команди натискання кнопок мишки зіставляються з різними конфігураціями руки.

с. Розпізнані позиції кінчиків пальців руки можуть бути використані для

управління комп'ютером або побутових приладів одночасно кількома пальцями руки.

2. Створення природних людино-машинних інтерфейсів для глухонімих

а. Систему розпізнавання жестів можна застосувати для введення тексту в комп'ютер за допомогою жестів руки, що для глухонімих людей простіше і природніше, ніж введення тексту за допомогою клавіатури комп'ютера.

б. Систему розпізнавання жестів можна застосувати для створення додатків онлайн зв'язку, коли пропускна здатність мережі мала і відеозв'язок неможливий.

3. Маніпуляція тривимірними моделями об'єктів

На сьогоднішній день, для роботи з тривимірними моделями зазвичай використовують комп'ютерну мишу, яка не дуже зручна для цього завдання. Маючи тривимірні координати руки і кінчиків пальців руки, можна створити систему HCI, яка дозволить керувати моделями в усіх напрямках тривимірного простору.

4. Додатки віртуальної реальності

Доповнивши систему розпізнавання жестів пристроями, наприклад стереоскопічними окулярами, можна створити додатки віртуальної реальності, де користувач зможе «доторкатися» до віртуальних об'єктів.

Наведений список можна доповнити системами, супутніми реабілітації пацієнтів [18], ігровими додатками [19] і т.д. Частина наведених додатків вже використовується в різних пристроях і програмах [18-20], а частина з них ще знаходиться на етапі досліджень.

Основні завдання розробки системи:

- аналіз існуючих методів побудови систем розпізнавання жестів рук людини;
- вибір найбільш відповідної апаратної бази;
- вибір засобів розробки програмного забезпечення і створення людино-машинного інтерфейсу.

Основними функціями системи є:

- розпізнавання та класифікація жестів людини в відеопотоці;
- виведення результатів класифікації на екран користувача;
- забезпечення користувача зручним графічним інтерфейсом, який може працювати як на комп'ютері, так і на мобільному пристрої;

— створити музичний плеєр що буде керуватися за допомогою жестів рук людини.

Висновки до розділу 1

Система розпізнавання жестів рук - це сукупність комп'ютерних технологій і математичних алгоритмів, яка дозволяє вирішувати задачу розпізнавання певної групи жестів руки. Дані можливості надають змогу автоматизувати сфери людської діяльності без використання клавіатури чи миши.

2. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗПІЗНАВАННЯ ЖЕСТІВ РУК

Особливістю методів, заснованих на аналізі зовнішніх ознак жесту, є аналіз тільки зовнішнього вигляду (форми, позиції і т.д.) цільового об'єкта. Для розпізнавання не зберігається ніякої інформації про фізичних властивості даного об'єкту.

Розглянемо відомі роботи і методи, присвячені розпізнаванню жестів руки людини на основі аналізу зовнішніх ознак жесту.

2.1 Опис сенсору LeapMotion

The Leap - це невеликий USB-пристрій, розроблений для користувачів, робочою частиною розташовується вгору, тим самим створюючи 3D-область взаємодії об'ємом близько 227 дециметрів кубічних (тобто уявний куб зі стороною 61 см). Як представлено на відеозаписах, всередині цієї області The Leap відстежує рух пальців і рук, олівців, ручок, паличок для їжі з великою точністю. [5]

На демонстрації для CNET, The Leap використовувався для навігації по веб-сайтам, збільшення карт за допомогою руху двох пальців, досить точного малювання і маніпуляцій з 3D об'єктами. [6]

На рисунку 2.1 зображено сенсор Leap Motion.



Рисунок 2.1 – LeapMotion сенсор

2.2 Опис браслету Myo

Myo Gesture Control складається з 8 секцій, що зчитують електричну активність імпульсів, що виникають при скороченні м'язів, щоб надати можливість управляти технологією за допомогою рухів руки. Для визначення активності браслета на ньому розташований світлодіод, який є індикатором. Секції з'єднуються еластичними кріпленнями, що дозволяють використовувати гаджет на різну ширину передпліччя.



Рисунок 2.2 – браслет Myo

2.3 Опис сенсору Kinect

Kinect (раніше *Project Natal*) — безконтактний сенсорний ігровий контролер, спочатку представлений для консолі Xbox 360, і значно пізніше для персональних комп'ютерів під керуванням ОС Windows. Розроблений фірмою Microsoft. Заснований на додаванні периферійного пристрою до гральної консолі Xbox 360, Kinect дозволяє користувачеві взаємодіяти з нею без допомоги контактного ігрового контролера через усні команди, пози тіла, об'єкти або малюнки. Мета проекту — збільшити число користувачів Xbox 360. Kinect для консолі Xbox був вперше представлений 1 червня 2009 року на виставці E3. У той же день фірма Microsoft розіслала розробникам програм понад 1000 комплектів засобів розробки Kinect. Kinect вийшов для консолі 4 листопада 2010 року в США і 10 листопада в Європі.

Поставки версії для Windows були розпочаті 1 лютого 2012 року.[5]

Проект Natal заснований на програмному забезпеченні розробленому корпорацією «Майкрософт» та веб-камери ZCam випущеної в 2000 р. фірмою 3DV Systems. ZCam є різновидом TOF-камери, що дозволяє отримувати тривимірну відео інформацію.



Рисунок 2.3 –Сенсор Kinect

2.4 Розпізнавання рухів на основі аналізу різниць зображень (MEI)

Використання різниць кадрів відеоряду (MEI) дозволяє в реальному часу аналізувати рух об'єкта в відеоряді при стабільному, але необов'язково однорідному зображенні. На практиці дана технологія та її модифікації (як приклад, motion history image - МНІ), знайшли застосування в таких додатках як інтерактивний віртуальний тренер з аеробіки і інтерактивна кімната для розповіді історій.

2.5 Розпізнавання конфігурації на основі аналізу гістограм напрямків

Більшості додатків комп'ютерного зору, крім позиції і орієнтації руки людини, потрібна додаткова інформація про її параметри. Дана проблема вирішується за допомогою так званих гістограм напрямків (orientation histograms) і карт напрямків (orientation maps) зображення, які не є чутливими до змін освітлення середовища.

Дані жести і дана технологія розпізнавання конфігурацій руки на основі гістограм напрямків, були використані для розробки додатка управління анімаційним краном. Додаток працює в реальному часі, показує гарні результати при незначних змінах розміру руки, але відчутний до змін руху руки. Додаток вимагає окремого додаткового навчання для кожної людини, що буде працювати з даним додатком.

Ця технологія дозволяє в режимі реального часу розпізнавати жест людини, якщо виконуються наступні умови:

1. Рука повинна займати більшу частину зображення.
2. На зображенні використовується однорідний фон.
3. Жести мають бути обрані таким чином, щоб гістограми напрямків мали значні відмінності один від одного.

2.6 Розпізнавання конфігурації і позиції із застосуванням кольорових рукавичок

Для вирішення задачі розпізнавання жестів руки часто застосовують кольорові рукавички. Даний метод дозволяє за допомогою встроєної відеокамери в реальному часі розпізнавати жести рук і відстежувати рухи долоні в тривимірному просторі. Рукавичка повинна бути сконструйованою з двадцяти сегментів десяти різних кольорів. Використання невеликої кількості кольорів дозволяє розпізнати колір обраної точки зображення рукавички при різному освітленні, а спеціальне розміщення колірних сегментів не дозволяє отримати ідентичні зображення при

різних конфігураціях руки.

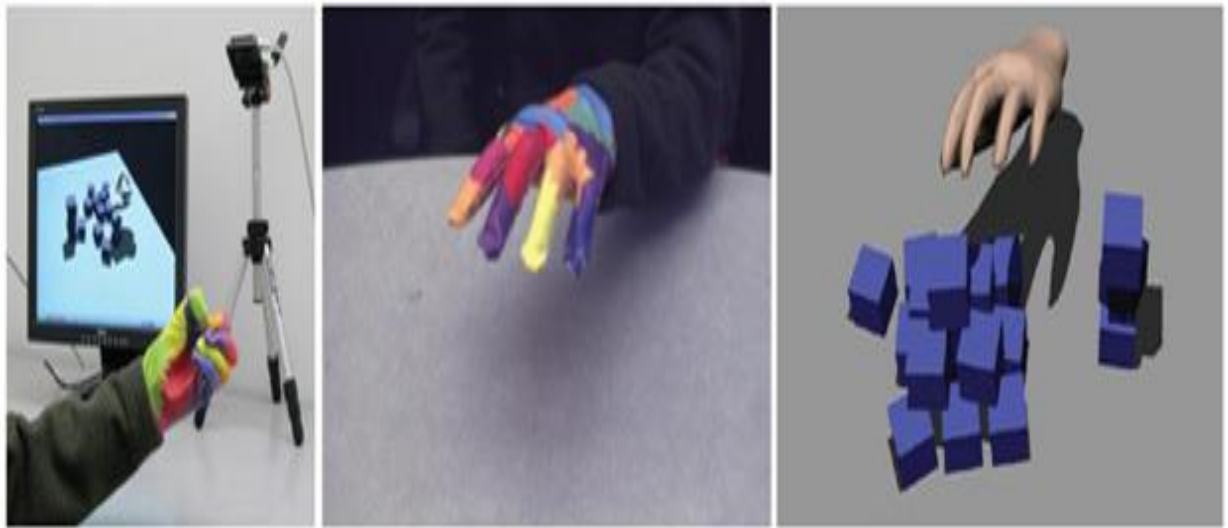


Рисунок 2.4 – Демонстрація роботи кольорових рукавичок

Дана система знайшла застосування в задачах управління анімаційними персонажами і розпізнавання жестів ручної азбуки глухонімих ASL.

2.7 Розпізнавання конфігурації і позиції на основі аналізу контуру зображення руки

В якості дескриптора конфігурації руки часто вибирають контур зображення руки. Наприклад, в роботі [33] пропонується система розпізнавання жестів ручної азбуки глухонімих ASL на основі аналізу контуру зображення руки. Для полегшення завдання два динамічних жестів розглядаються, і частина жестів замінюється новими. У кольоровому зображенні руки видаляються всі точки, які не відповідають кольору шкіри людини. Отримане зображення перетворюється в бінарне зображення, після чого згладжується фільтром Гауса. Контури руки виділяються за допомогою оператора Собеля. (Оператор Собеля використовується в обробці зображень для виділення границь. Це дискретний диференціальний оператор, що обчислює наближене значення градієнта чи норми градієнта для яскравості зображення. Оператор Собеля базується на згортці зображення невеликими сепарабельними цілочисельними фільтрами в вертикальному та горизонтальному

напрямках.)

Обчислюється описаний прямокутник контуру руки, центр якого відповідає початку координат.

В роботі [34] розглядається задача розпізнавання позиції кінчиків пальців руки на основі аналізу контуру руки в кольоровому зображенні. Першими кроками алгоритму, як і в випадку [33], є видалення фонових точок, згладжування і виділення зображення руки. Маючи контур руки, пальці руки виділяються за допомогою аналізу локальних вигинів контуру. Даний алгоритм використовувався для управління веб камерою персонального комп'ютера.

На рисунку 2.5 показан приклад пошуку контурів руки людини. На малюнку ліворуч – рука з видаленим фоном, праворук – показана реалізація цього алгоритму. Проблема цього алгоритма: розпізнаються лише прості жести, із-за наявності фону жест може бути не розпізнаним. На роботу алгоритма впливає і освітлення, і складність фону. Метод рідко використовується самостійно, найчастіше цей метод включають у процес попередньої обробки зображення(препроцесинг).

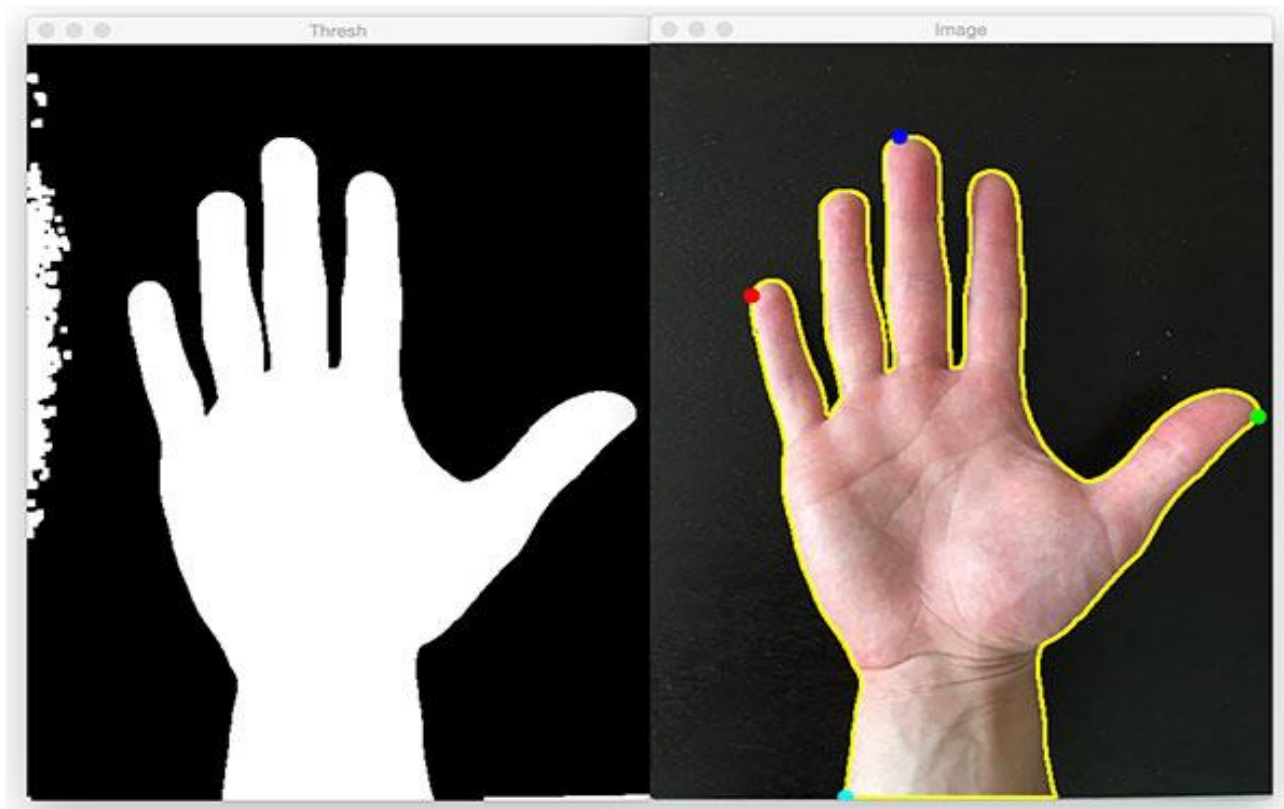


Рисунок 2.5 – Пошук контурів руки

2.8 Розпізнавання жестів руки із застосуванням штучних нейронних мереж

Штучні нейронні мережі (ШНМ) в своїй сутності є математичною моделлю, що побудована на принципах організації і функціонування біологічних нейронних мереж. Первинною метою підходу ШНМ було розв'язання задач таким же способом, як це робив би людський мозок. З часом увага зосередилася на відповідності певним розумовим здібностям, ведучи до відхилень від біології. ШНМ використовували в ряді різноманітних задач, включно з комп'ютерним баченням, розпізнаванням мовлення, машинним перекладом, соціально-мережовим фільтруванням, грою в настільні та відеоігри, та медичним діагностуванням. У завданнях розпізнавання жестів рук людини, ШНС як правило використовують в якості інструменту машинного навчання. На вхід ШНС передаються математичні характеристики жесту і на виході виходить номер розпізаного жесту. ШНС відрізняються між собою структурою, різноманітними моделями та методами навчання. Найбільш поширеними видами ШНС, що використовуються в задачах розпізнавання, є мережі прямого поширення (англ. Feed forward neural networks), рекурентні нейронні мережі, карти Кохонена і, згорткові нейронні мережі.

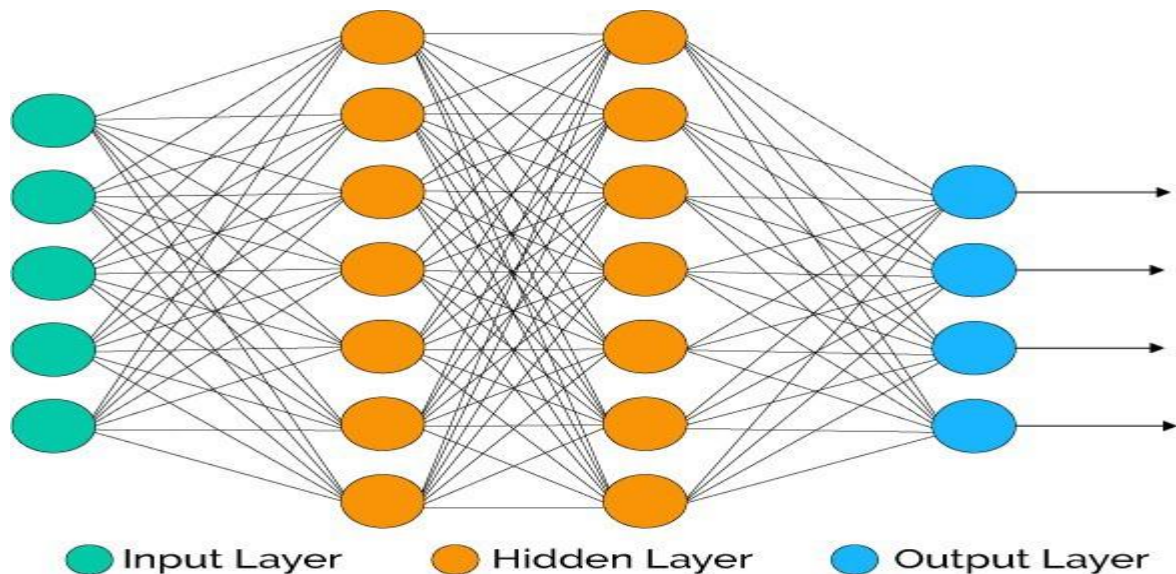


Рисунок 2.6 – Схема згорткової нейронної мережі

В Японії була запропонована рекурентна нейронна мережа для розпізнавання статичних жестів для азбуки глухонімих. В якості вхідного пристрою в роботі використовувалась рукавичка з вбудованими сенсорами, яка надає інформацію про характеристики показаного жесту (10 точок і 3 кута). Кожна характеристика жесту нормалізується і передається на вхід нейронної мережі.

Дана мережа повинна була видавати результат для 42 різних жестів. Кінцеві результати розпізнавання - 71.4% для користувачів на жестах яких відбувалося навчання, і 47.8% для нових користувачів.

2.9 Аналіз існуючих методів розпізнавання жестів руки на основі аналізу тривимірної моделі руки

Дана технологія розпізнавання жестів використовується в комп'ютерному зорі для отримання детальної тривимірної конфігурації руки при наявності на вході одного або декількох двовірних зображень жесту. Позиція і орієнтація долоні і ключових точок пальців руки надається в тривимірному просторі.

Жести розпізнаються за допомогою порівняння проєкцій тривимірної моделі руки з двовірним вхідними зображеннями. Тривимірна модель руки проєктується на площину, тим самим, виходить двовірне зображення, яке порівнюється з вхідним зображенням. Якщо відстань між зображеннями менше заданого порогу, то жест розпізнано, в іншому випадку конфігурація моделі руки змінюється і заново порівнюється з вхідним зображенням. Гіпотези, що використовуються в даних системах різні, але в більшості випадків передбачається, що конфігурація руки в кожному кадрі відеоряду збігається або незначно відрізняється від знайденої конфігурації руки в попередньому кадрі. Головним недоліком цих методів є потреба у великих ресурсах. Розпізнавання жесту в одному кадрі іноді вимагає навіть 15 секунд для виконання, або 66 мілісекунди при використанні ресурсів графічного процесору. Саме тому у багатьох роботах вводяться обмеження на кількість точок артикуляції.

Висновки до розділу 2

Таким чином, існуючі методи можна розділити на 3 основні групи:

1. Методи, засновані на використанні відеокамер;
2. Методи, засновані на використанні спеціальних рукавичок, що оснащені сенсорами;
3. Методи, засновані на використанні тривимірних сенсорів.

Перша група методів дозволяє розпізнавати статичні і динамічні жести руки, і в деяких випадках з точністю розпізнавання більше 90% [41, 54, 56]. Недоліком цих методів є в основному чутливість до змін освітлення, яку в різних роботах намагаються усунути за допомогою використання кольорових рукавичок або створенням однорідного фону, що робить людино-машинну взаємодію незручним і неприродним.

Методи другої групи дозволяють з високою якістю розпізнавати позиції ключових точок руки людини, але натомість вимагають використання спеціальних рукавичок.

Методи третьої групи дозволяють в реальному часі розпізнавати ключові точки тіла людини, є незалежними від зміни освітлення, але на сьогоднішній день не існує готових рішень, що дозволяють з великою точністю розпізнавати конфігурації і динамічні жести руки людини.

3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Аналізуючи поставлену задачу та методи її розв'язку, було вирішено розроблювати програмний комплекс в якості десктопної програми - програми, логіка роботи якої вимагає наявності оператора (людини що працює з програмою), що містить в собі всю повну функціональність і здатна працювати окремо на будь-якій машині ізольовано від інших додатків.

У даному розділі описано особливості побудови та навчання нейронної мережі. Для побудови моделі для нейронної системи використовувалися кращі практики теорії розв'язання винахідницьких задач (ТРВЗ).

3.1 Особливості побудови моделі для нейронної мережі з використанням теорії розв'язання винахідницьких задач

Під час проектування системи було проведено декілька експериментів для вибору найбільш підходящої моделі, класифікатора та найбільш підходящої функції втрат.

Для зручності розробки був додатково розроблений модуль для зберігання та зчитування даних, а також модуль, завдяки якому задавалися додаткові параметри для нейронної мережі що навчається (кількість епох, ймовірність дропаута, кількість шарів та ін.).

Для навчання використовувався датасет, наданий Kaggle, що складається з більш ніж 6000 фотографій, зроблені за допомогою LeapMotion камери.

При виборі найбільш відповідної моделі нейронної мережі використовувалися кращі практики ТРВЗ (теорії розв'язання винахідницьких задач).

Різні списки винахідницьких прийомів з початку XX-го століття публікувалися неодноразово. Автори публікацій доволі часто включали в них ті прийоми, які здавалися їм найкращими, не замислюючись над природою цих прийомів.

Технічним (системним) протиріччям в ТРВЗ називається ситуація, коли спроба поліпшити одну характеристику технічної системи викликає погіршення іншого[20].

Аналіз великих масивів патентної інформації показав, що для усунення приблизно півтори тисячі технічних протиріч знадобилось всього лише 40 найбільш сильних прийомів, що дають ефективні рішення. У завданні розпізнавання жестів були знайдені декілька технічних протиріч, та запропоновані їх вирішення за допомогою ТРВЗ[21].

Отже, існує проблема: необхідно створити модель для нейронної мережі, яка дозволить розпізнавати жести рук людини. Найважливіші проблеми, що виникають при розробці – наявність фону, недостатня кількість фотографій(семплів), проблема розпізнавання жестів лише для однієї людини.

Для вирішення проблем були обрані наступні пункти з принципів вирішення технічних протиріч: принцип асиметрії, принцип винесення, принцип переходу в інший вимір, принцип об'єднання.

Принцип асиметрії - цей принцип говорить про те, що для вирішення деяких протиріч необхідно:

- а) перейти від симетричної форми об'єкта до асиметричної;
- б) якщо об'єкт асиметричний, збільшити ступінь асиметрії.

В нейронних мережах такий підхід використовується скрізь при створенні датасету для навчання. Цей метод допоможе нам вирішити технічне протиріччя - розпізнавання жестів лише для одного оператора. Основна проблема (технічне протиріччя) – жести повинні розпізнаватися для різних людей(операторів), але при цьому не бажано використовувати великий датасет.

Отже, для розв'язання задачі необхідно додати трансформації, які будуть спотворювати вихідне зображення. Для цього скористуємось існуючим інструментом, що має назву у pytorch – transform.

```

self.transformations = transforms.Compose([
    transforms.Resize(255),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])

```

Рисунок 3.1 – Трансформації зображення

Жести руки - складне зображення. Зображення, що містить як жести рук людини, так і саму людину, займає велике місце в пам'яті комп'ютера, що уповільнює роботу нейронної мережі.

Наступне технічне протиріччя: для розпізнання жестів необхідна велика кількість даних, що зчитуються з зображення, в той же час із-за великої кількості даних, що представлені на зображенні, розпізнавання жестів сповільнюється.

Для вирішення даної проблеми, повертає увагу наступний принцип - принцип винесення: відокремити від об'єкта "непотрібну" частину ("непотрібну" властивість) або, навпаки, виділити єдину потрібну частину (потрібну властивість).

Цей принцип привернув свою увагу і розробників нейронних мереж, і навіть отримав власну назву – субдискретизація. Операція субдискретизації (англ. Subsampling, англ. Pooling, також перекладається як «операція підвибірки» або операція об'єднання), виконує зменшення розмірності сформованих карт ознак. У даній архітектурі мережі вважається, що інформація про факт наявності шуканої ознаки важливіше точного знання його координат, тому з кількох сусідніх нейронів карти ознак вибирається максимальний і приймається за один нейрон ущільненої карти ознак меншої розмірності. За рахунок цієї операції, крім прискорення подальших обчислень, мережа стає більш інваріантною до масштабу вхідного зображення.

Pytorch має необхідні інструменти для реалізації - наприклад MaxPool2d, - карта ознак поділяється на клітинки розмірністю 2x2 елемента, з яких вибираються максимальні значення. Формально шар може бути описаний таким чином.

$$x^l = f(a^l \cdot \text{subsample}(x^{l-1}) + b^l) \quad (3.1)$$

Де x_l - вихід шару l ;

$f()$ - функція активації;

a, b – коефіцієнти;

$\text{subsample}()$ - операція вибірки локальних максимальних значень.

Використання цього шару дозволяє поліпшити розпізнавання зразків зі зміненим масштабом (зменшених або збільшених). Більш того цей шар дозволяє економити пам'ять, прискорює навчання.

Окрім цього обраний датасет частково вирішує проблему «непотрібної частини» завдяки тому, що на фотографіях з датасету відсутній фон.

Цей принцип знайшов своє місце також і в шарах згортки, де ми обираємо ознаки для розпізнавання зображення. Pytorch надає можливість виконати цю вимогу за допомогою методу **Conv2d**. Для зручності експериментів, було створено метод **conv3x3**.

```
def conv3x3(in_planes, out_planes, stride=1, groups=1, dilation=1, padding = 1):
    """3x3 convolution with padding"""
    return nn.Conv2d(in_planes, out_planes, kernel_size=3, stride=stride, padding=padding, groups=groups, bias=False, dilation=dilation)
```

Рисунок 3.2 – Шар згортки

Для власних експериментів, щоб виконати вимоги принципу асиметрії, було створено додаткові методи, зі власними ядрами – з гаусовим викривленням, ядром для пошуку контурів, загостренням зображення, а також зі власним ядром.

Нижче наведений приклад створювання ядра з гаусовим викривленням:

```
def get_gaussian_kernel(kernel_size=3, sigma=2, channels=3):
    x_coord = torch.arange(kernel_size)
    x_grid = x_coord.repeat(kernel_size).view(kernel_size, kernel_size)
    y_grid = x_grid.t()
    xy_grid = torch.stack([x_grid, y_grid], dim=-1).float()
    mean = (kernel_size - 1) / 2.
    variance = sigma ** 2.
    gaussian_kernel = (1. / (2. * pi * variance)) * \
        torch.exp(
            -torch.sum((xy_grid - mean) ** 2., dim=-1) / \
            (2 * variance)
        )
    gaussian_kernel = gaussian_kernel / torch.sum(gaussian_kernel)
    gaussian_kernel = gaussian_kernel.view(1, 1, kernel_size, kernel_size)
    gaussian_kernel = gaussian_kernel.repeat(channels, 1, 1, 1)
    gaussian_filter = nn.Conv2d(in_channels=channels, out_channels=channels,
                                kernel_size=kernel_size, groups=channels, bias=False)
    gaussian_filter.weight.data = gaussian_kernel
    gaussian_filter.weight.requires_grad = False
```

Рисунок 3.3 – Ядро з гаусовим викривленням

Такі ж методи були створені для інших ядер:

```
outline_kernel - weights = torch.Tensor([[[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]]).unsqueeze(0).unsqueeze(0)
sharpen_kernel - weights = torch.Tensor([[0, -1, 0], [-1, 5, -1], [0, -1, 0]]).unsqueeze(0).unsqueeze(0)
custom_kernel - weights = torch.Tensor([[[-1, 0, 1], [0, 0, 0], [1, 0, -1]]]).unsqueeze(0).unsqueeze(0)
```

Рисунок 3.4 – Створення власних ядер з викривленнями

Одним з найбільш цікавих принципів для побудови моделі нейронної мережі є принцип переходу в інший вимір:

- а) труднощі, пов'язані з рухом (або розміщенням) об'єкта по лінії, усуваються, якщо об'єкт набуває можливість переміщатися в двох вимірах (на площині). Відповідно завдання, пов'язані з рухом (або розміщенням) об'єктів в одній площині, усуваються при переході до простору в трьох вимірах;
- б) використовувати багатоповерхову компоновку об'єктів замість одноповерхової.

Над цим завданням (проеціювання 2Д зображення в 3Д зображення) працюють безліч дослідників по всьому світу. Однак вирішити цю задачу до сих пір не вдається.

Людина вирішує подібні завдання за допомогою просторового мислення - однієї з важливих складових інтелекту. З її допомогою ми можемо орієнтуватися в просторі, вирішувати геометричні завдання, представляти предмети в тривимірному

просторі. Іншими словами, просторове мислення - це здатність людини уявити об'єкт у всіх його деталях і проявах і будь-яким чином трансформувати цей об'єкт.

Роботи Джона О'Кіфа (John O'Keefe), Мей-Брітт Мозер (May-Britt Moser) і Едварда Мозера (Edvard Moser) у нейробиології допомогли описати вроджену систему орієнтації в просторі. За свої дослідження вчені удостоїлися Нобелівської премії з фізіології і медицині в 2014 році «за відкриття клітин системи позиціонування мозку» - половину премії присудили О'Кіф, а половину - подружжю Мозер.

Однак їх роботи описують лише детально як працює нейронна система при вирішенні просторових задач. Навіть сьогодні робота просторового мислення у людей для вчених усього світу представляється загадкою.

Оскільки нейронні мережі в системах штучного інтелекту були побудовані за тим же принципом, що і нейронна мережа головного мозку, то очікувати відкриття в сфері штучного інтелекту в цьому напрямку поки не варто. Однак Г.Альтшуллер представив цей принцип вирішення технічних протиріч, який через кілька десятків або навіть сотень років, знайде собі застосування і в штучному інтелекті.

Останній принцип, що привертає увагу – принцип об'єднання:

- а) з'єднати однорідні або призначені для суміжних операцій об'єкти;
- б) об'єднати у часі однорідні або суміжні операції.

У нейронних мережах цей принцип також знайшов своє застосування у так званому повнозв'язному шарі (Fully-connected) – прихований шар штучної нейронної мережі типу персептрон.

Основне призначення даного шару - перетворення сигналів, отриманих на попередніх рівнях мережі до одновимірного виду, а також виділення ознак на одновимірному рівні. Даний шар також може використовуватися в якості останнього (вихідного) шару нейронної мережі, результатом якого є ймовірність приналежності вхідного зображення до певного класу.

У власних експериментах було створено декілька варіантів повнозв'язного шару.

Нижче наведені архітектури створених моделей.

Перша модель - модель складається з одного шару. У побудові моделі

використовувався тільки принцип об'єднання.

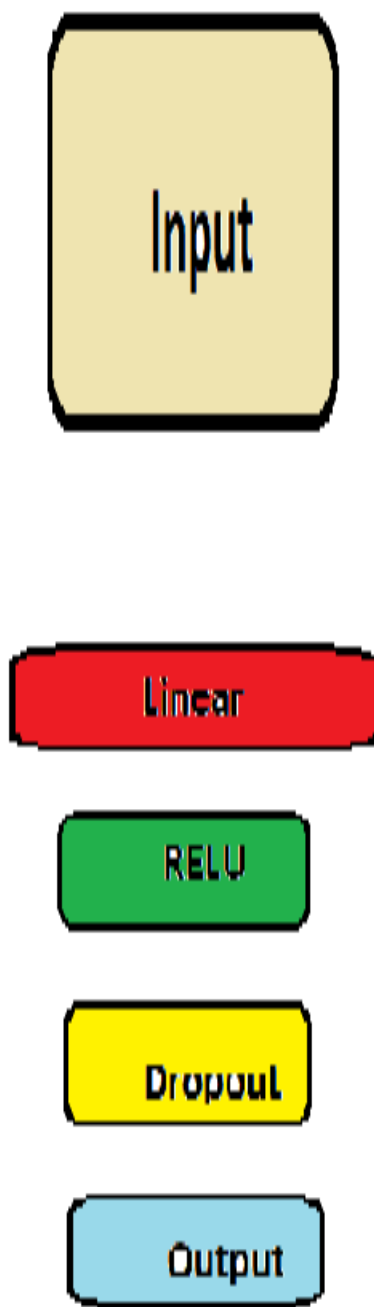


Рисунок 3.5 – Нейронна мережа що складається з одного шару

Результати роботи показані на малюнку 3.6

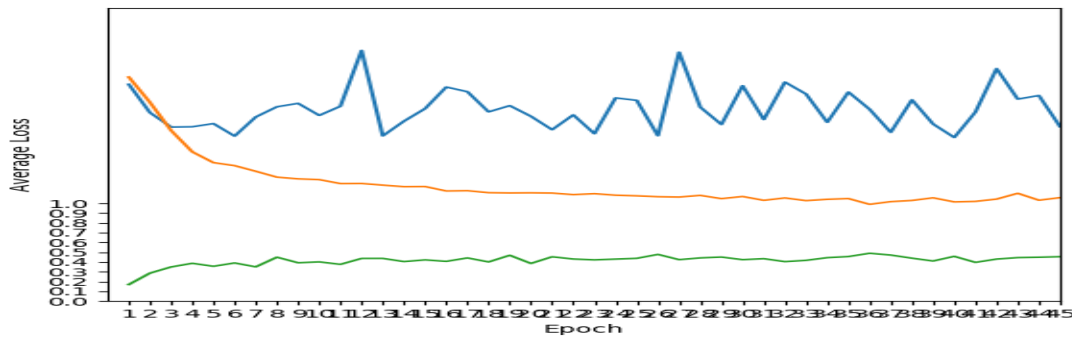


Рисунок 3.6 – Результати роботи неройнної мережі що складається з єдиного шару

Як бачимо, за 45 епох модель показала найкращі результати на епосі номер 18 - 46.848958 відсотків точності та припинила навчання.

Згідно з ТРВЗ, якщо можливості розвитку, тобто можливості внутрішньої перебудови системи, вичерпуються, то система змінюється для включення в надсистему як складова частина або для виконання функцій надсистеми; після цього подальший розвиток системи йде в складі і на рівні надсистеми.

Згідно ТРВЗ, перехід в надсистему шляхом об'єднання йде по шляху «моно-бі-полі» і починається з об'єднання двох (або декількох) вихідних систем. Бі-систему отримують, об'єднуючи для якоїсь мети дві однакові (або майже однакові) моно-системи. Об'єднуючи - значить, встановлюючи якийсь зв'язок. Полі-систему отримують об'єднанням декількох моно-систем. При такому об'єднанні кожна моно-система зберігає свої властивості, а зв'язок в сукупності з властивостями моно-систем допомагає новій системі (по суті - надсистемі) виконувати нову функцію, що недоступна моно-системам. Іншими словами, якась потреба суспільства (функція ієрархії систем) може бути виконана ефективніше, ніж раніше, якщо об'єднати кілька моно-систем.[22]

Об'єднувати можна:

- однорідні (однофункціональні) системи,
- системи з зсунутими (що трохи відрізняються) характеристиками,
- різнорідні (мають різні функції) системи,
- інверсні (виконують протилежні функції) системи.

Об'єднання і злиття систем в усіх зазначених випадках проходить в одні й ті ж етапи.

Перехід моно-бі-полі здійснюється на будь-якому етапі розвитку і виконується для всіх рівнів ієрархії.

Створимо полісистему для вирішення завдання розпізнавання жестів.

Друга модель - модель складається з чотирьох шарів. Архітектура моделі показана на рисунку 3.7.

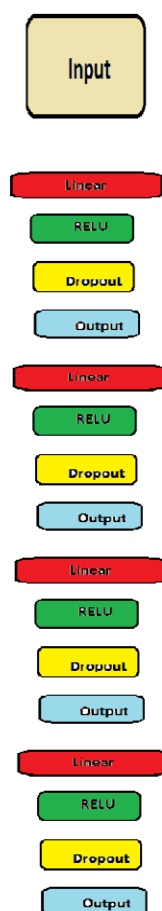


Рисунок 3.7 – Архітектура нейронної мережі що складається з чотирьох шарів

Як бачимо, модель дуже проста, складається з чотирьох шарів. В якості активації використовує функцію RELU.

Результати навчання цієї моделі показані на малюнку 3.8.

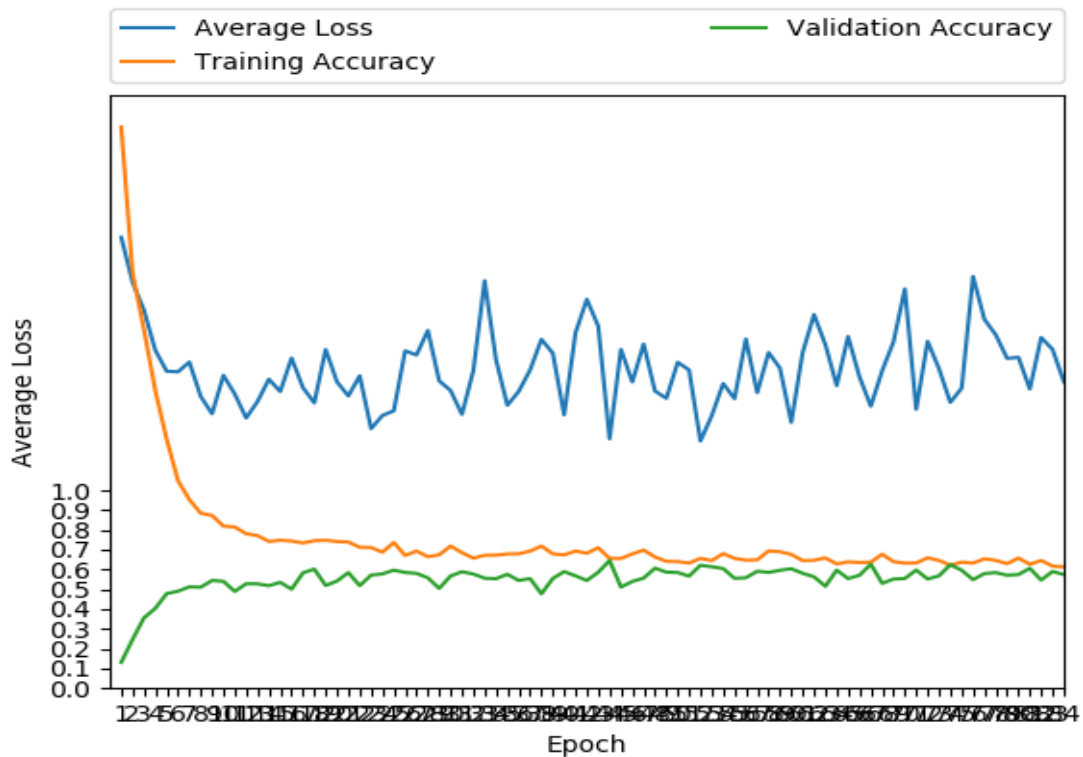


Рисунок 3.8 – Результати роботи моделі що складається з чотирьох шарів

Найкращі показники для цієї моделі показала епоха номер 66 з точністю 62.656250 відсотків. Полісистема показала значно кращі результати.

Наступні етапи в створенні системи – використання принципів асиметрії та принцип винесення.

Для цього для датасету були застосовані трансформації, була ускладнена вся система за допомогою додавання згорткових шарів, шарів нормалізації та пулінга. Також експерименти показали, що модель показує кращі результати якщо у шарі згортки згортку провести зі своїм ядром. Найкращі результати показало ядро, що називається ядром Гауссіани, та використовується для розмиття зображення.

Архітектура системи показана на наступному рисунку(3.9).

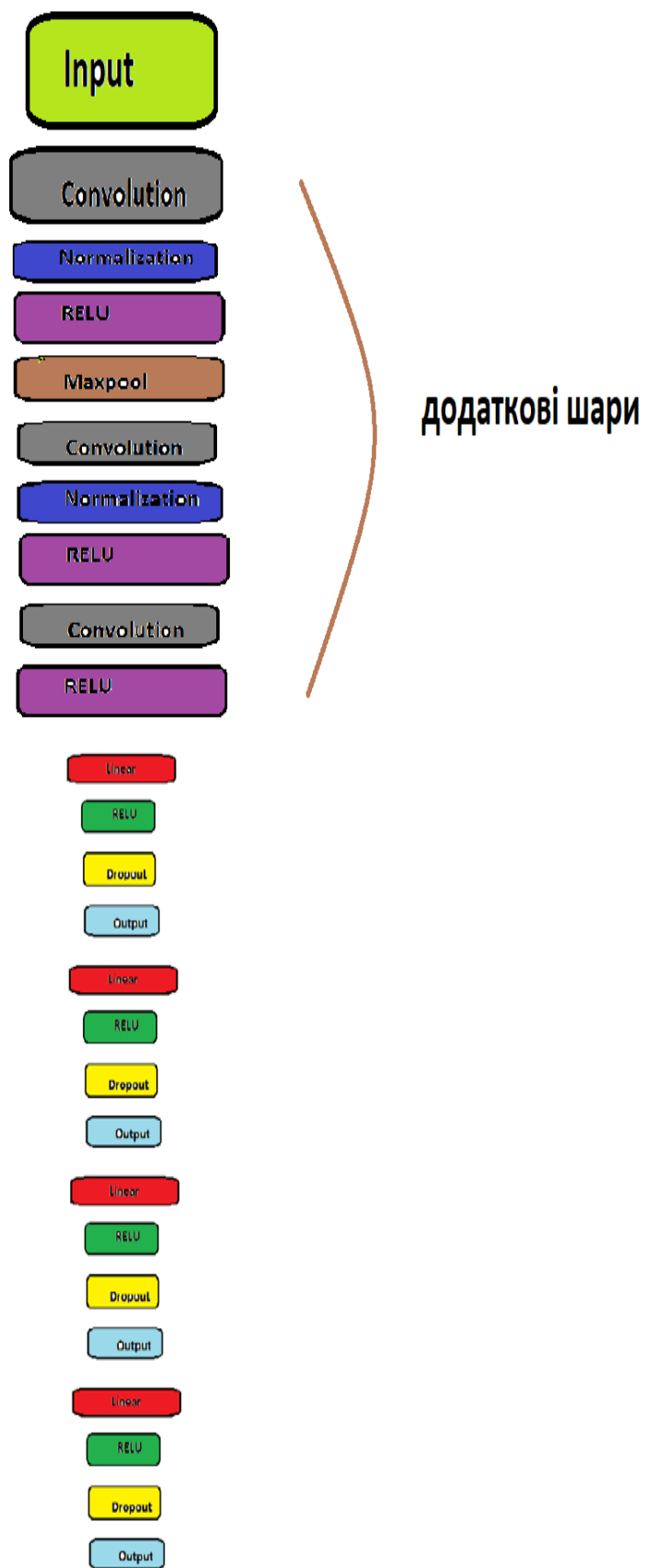


Рисунок 3.9 – Архітектура моделі з додатковими шарами

Результати роботи цієї моделі показали результати, показані на рисунку 3.10.

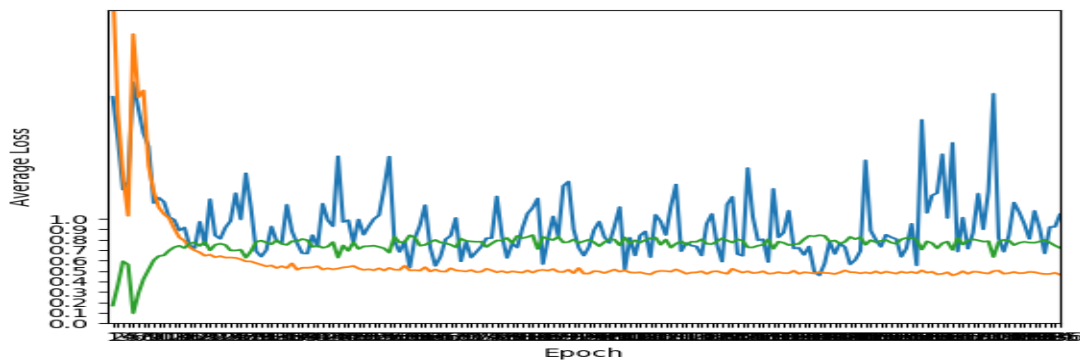


Рисунок 3.10 – Результати роботи моделі

Найкращі показники показала епоха 76 - 92.375000 %.

Остання модель показала найкращі результати. Тому для подальшої розробки було прийняте рішення використовувати цю модель.

3.2 Опис інструментів розробки

Головною метою було створення мультиплатформного рішення з використанням відкритих технологій.

При розробці даної системи використовувались найактуальніші технології та методики, що застосовуються у ІТ галузі на сьогоднішній день. Значний акцент було зроблено на досягнення відповідності до керівних вказівок від провідних компаній світу – таких як Microsoft, Google, Facebook.

Розроблена система складається з трьох модулів – модуль навчання, модуль виведення інформації про результати навчання та модуль з музикальним плеєром. Тому розглянемо особливості інструментів, що використовувалися в розробці цих модулів.

Для розробки модуля навчання використовувалась мова програмування Python, бібліотека комп'ютерного зору openCV, бібліотека для роботи з нейронними мережами pytorch. Щоб прискорити роботу додатку, було прийняте рішення зібрати

бібліотеку `openCV` власноруч за допомогою інструменту `CMake` – це надало можливість проводити обчислення на GPU, що прискорило роботу додатку в декілька разів[24].

У даний час спостерігається стрімкий розвиток персональної обчислювальної техніки, відбувається поступова зміна вимог, що пред'являються до мов програмування. Інтерпретовані мови починають грати все більшу роль, оскільки зростаюча міць персональних комп'ютерів починає забезпечувати достатню швидкість виконання інтерпретованих програм. А єдиною суттєвою перевагою компільованих мов програмування є створюваний ними високошвидкісний код. Коли швидкість виконання програми не є критичною величиною, хорошим вибором буде інтерпретована мова, як більш простий і гнучкий інструмент програмування, що дозволяє отримати швидкі результати.

У зв'язку з цим, в даний час достатньо поширеною є мова Python (пайтон), яка була створена Гвідо ван Россум (Guido van Rossum) на початку 90-х років.

Переваги мови:

- Python реалізована практично на всіх платформах і операційних системах. Першою такою мовою була мова C, проте типи даних у мові програмування C на різних машинах можуть займати різну кількість пам'яті - це є деякою перешкодою при написанні платформонезалежних програм. Python ж таким недоліком не володіє.

- Розширюваність мови. Цьому приділяється велика увага і, як пише сам автор, мову було задумано саме як розширювану мову. Це означає, що кожний зацікавлений програміст має можливість вдосконалення мови. Інтерпретатор у Python написаний на C і вихідний код доступний для будь-яких маніпуляцій. У разі необхідності, мови C та C++ можна вставити в свою програму, написану на Python і використовувати як вбудовану оболонку. Або ж, написавши на C свої доповнення до Python і скомпільовавши програму, отримати "розширений" інтерпретатор з новими можливостями.

- Наявність великого числа підключених до програми модулів, що забезпечують додаткові можливості. Такі модулі пишуться на C і на самому Python і можуть бути розроблені усіма досить кваліфікованими програмістами. Як приклад

можна навести такі модулі:

Numerical Python - розширені математичні можливості, такі як маніпуляції з цілими векторами і матрицями;

Tkinter - побудова додатків з використанням графічного інтерфейсу користувача (GUI) на основі широко розповсюдженого на X-Windows Tk-інтерфейсу;

OpenGL - використання великої бібліотеки графічного моделювання дво- і тривимірних об'єктів Open Graphics Library фірми Silicon Graphics Inc. Даний стандарт підтримується, в тому числі, в таких поширених операційних системах як Microsoft Windows 95 OSR 2, 98 і Windows NT 4.0.

Недоліки мови:

Єдиним недоліком є порівняно невисока швидкість виконання Python-програми, що обумовлено тим, що мова є інтерпретованою.

Мова програмування Python дуже популярна в сфері розробки нейронних мереж завдяки простоті та зручності. Саме тому саме для мови Python було розроблено декілька бібліотек – pytorch, keras, tensorflow. Ці бібліотеки полегшують роботу з нейронними мережами, а також надають можливості для експериментів.

PyTorch - це бібліотека на Python, що забезпечує тензорні обчислення з GPU-прискоренням, подібно NumPy. Понад це PyTorch пропонує насичений API для вирішення прикладних завдань, пов'язаних з нейронними мережами[23].

PyTorch відрізняється від інших фреймворків машинного навчання тим, що тут не використовуються статичні розрахункові графи(визначаються заздалегідь, відразу і остаточно - як в TensorFlow, Caffe2 або MXNet). Навпаки, розрахункові графи в PyTorch динамічні і визначаються на льоту.

OpenCV (англ. *Open Source Computer Vision Library*, бібліотека комп'ютерного зору з відкритим кодом) — бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних

зображеннях.

Бібліотека розроблена Intel і нині підтримується Willow Garage та Itseez. Сирцевий код бібліотеки написаний мовою C++ і поширюється під ліцензією BSD. Біндинги підготовлені для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua та інших. Може вільно використовуватися в академічних та комерційних цілях.

Для збірки бібліотеки openCV використовувався CMake.

CMake — крос-платформовий відкритий генератор сценаріїв складання. CMake не займається безпосередньо складанням, а лише генерує файли управління складанням з файлів CMakeLists.txt[24]:

- Makefile в системах Unix для складання з допомогою make;
- файли projects/workspaces (.dsp/.dsw) у Windows для збірки з допомогою Visual C++;
- проекти XCode у Mac OS X.

З переваг CMake можна відзначити: проста мова сценаріїв, засоби розширення функціональності через модулі, мінімальне число залежностей (немає прив'язки до M4, Perl або Python), підтримка кешування, наявність засобів для крос-компіляції, підтримка генерації файлів складання для широкого спектра систем складання і компіляторів, наявність утиліт ctest і cpack для визначення сценаріїв тестування і складання пакунків, утиліта cmake-gui для інтерактивного налаштування параметрів складання.

CMake виступає як альтернатива Autotools і використовується в таких проектах, як KDE, LLVM/Clang, MySQL, MariaDB, ReactOS і Blender. Сирцевий код CMake написаний мовою C++ і поширюється під ліцензією BSD.

Для розробки модуля виводу інформації про навчання використовувалася мова програмування Python,

Для розробки модуля з музикальним плеєром використовувалася мова програмування Python, бібліотека vlc, бібліотека openCV, модуль розпізнавання мови. Для зберігання даних використовувався інструмент git.

Медіапрогравач VLC (англ. VideoLAN Client) - безкоштовний, портативний і

крос-платформний медіаплеєр з відкритим вихідним кодом, розроблений в рамках VideoLAN проекту. Даний медіаплеєр доступний для безлічі настільних операційних систем, таких як Windows, Linux, Mac OS. Також VLC доступний для мобільних операційних систем на кшталт Tizen, IOS, Android, основним способом поширення на даних платформах є цифрова дистрибуція (App Store від Apple, Google Play від Android і Windows Store).[25]

VLC підтримує безліч видів стиснення аудіо і відео, форматів файлів, включаючи DVD-Video, відео CD і потокові протоколи. Він здатний передавати мультимедійні файли по комп'ютерних мережах і розкодувати мультимедійні файли. За замовчуванням VLC включає в себе велику кількість вільних бібліотек декодування і кодування, уникаючи необхідності пошуку і калібрування власних плагінів. Хоч в основному VLC плеєр використовує власні мультиплексори і демультиплексори, він також використовує сторонні бібліотеки з інших проектів, таких як FFmpeg. VLC є першим програвачем з підтримкою зашифрованих DVD дисків на платформах Linux і Mac OS, цього вдалося досягти за допомогою бібліотеки розшифровки DVD libdvdcss.

3.3 Обґрунтування вибору програмної реалізації

При проектуванні системи було вивчено та проаналізовано предметну область та вимоги замовника. Після ретельного аналізу було вирішено розроблювати програмний продукт, який реалізований з використанням бібліотеки OpenCV, що була забрана для GPU за допомогою інструменту CMake, мови програмування Python, бібліотеки pytorch, бібліотеки vlc для обробки аудіо та медіа інформації.

Python - це мова програмування загального призначення, націлена в першу чергу на підвищення продуктивності програміста. На Python можна написати практично що завгодно (веб- / настільні додатки, ігри, скрипти з автоматизації, комплексні системи розрахунку, системи управління життєзабезпеченням і багато

іншого) без відчутних проблем. За рахунок простоти коду, подальший супровід програм, написаних на Python, стає легшим і приємнішим, а з точки зору бізнесу це тягне за собою скорочення витрат і збільшення продуктивності праці співробітників.[27]

Для створення моделі для нейронної мережі було обрано бібліотеку `pyTorch` - це науковий обчислювальний пакет на основі Python, який використовує потужності графічних процесорів. Це також одна з кращих дослідницьких платформ глибокого навчання, створена для забезпечення максимальної гнучкості і швидкості. Він відомий тим, що забезпечує дві з найбільш високорівневих функцій; а саме, тензорні обчислення з сильною підтримкою прискорення GPU і побудову глибоких нейронних мереж.[28]

Однією з ключових причин успіху `PyTorch` є те, що він повністю використовує мову Python, і надає можливості створювання власних моделей нейронних мереж.

Деякі з ключових моментів `PyTorch` включають:

- Простий інтерфейс: він пропонує простий у використанні API, тому він дуже простий в управлінні і працює як Python.
- Обчислювальні графіки: на додаток до цього `PyTorch` забезпечує відмінну платформу, яка пропонує динамічні обчислювальні графіки, тому їх можна змінити їх під час виконання.

Для препроцесингу було обрано використовувати бібліотеку `openCV` через те що вона написана на C, добре оптимізована і може використовувати переваги багатоядерних процесорів. `OpenCV`, встановлена за допомогою стандартних пакетів, працює з використанням CPU, що в кінцевому підсумку, позначиться на продуктивності. Саме тому було прийнято рішення зібрати дану бібліотеку для GPU власноруч з використанням вихідного коду та інструменту `cMake`.

Для забезпечення розпізнавання жестів людини було побудовано власну модель, а дані були отримані з датасету від Kaggle. Цей датасет було обрано тому, що він містить більше 6000 зображень для 15 різних жестів, зроблених за допомогою камери LeapMotion.

Для роботи з аудіо інформацією було обрано бібліотеку `vlc` через те, що це

проект з відкритим вихідним кодом і надзвичайно великою підтримкою з боку розробників. Також важливою властивістю є велика швидкість роботи, надійність та кількість вбудованих можливостей. Цю бібліотеку можливо використовувати на будь-якій сучасній операційній системі або в Docker контейнері.

Дані технології в сукупності дають змогу збудувати якісний та надійний продукт, який захищений від патентних позовів з боку розробників, бо всі ці технології покриті ліцензіями, які виключають таку можливість і надають доступ до вихідних кодів даних проектів.

3.4 Обґрунтування вибору згорткової нейронної мережі

Згорткова нейронна мережа (Convolutional Neural Network) – це архітектура штучної нейронної мережі, що імітує особливості зорової кори головного мозку. Складається з декількох багатовимірних шарів і призначена для ефективного розпізнавання складних зображень. Вперше така модель була запропонована Яном Лекуном (Yann Lecun) в 1998 році і призначалася для розпізнавання рукописних символів. Особливість запропонованої Лекуном моделі нейронної мережі полягала у введенні в архітектуру багатошарового персептрона згорткових шарів, в яких кожен нейрон був пов'язаний тільки з невеликою областю нейронів попереднього шару. Така особливість дозволяла виділяти на оригінальному документі примітивні особливості, а на наступних шарах мережі, об'єднуючи виділені ознаки, отримувати всі складніші елементи, таким чином виявляючи складні об'єкти на зображеннях.

Під час створення першої ЗНМ обчислювальні потужності комп'ютера були порівняно малими, а накопичених на той час візуальних даних було недостатньо для того, щоб такі математичні моделі отримали широке поширення. В даний час ЗНМ і машинне навчання є одними з головних напрямків розвитку більшості сфер, пов'язаних з інформаційними технологіями.

Однак базові принципи побудови архітектури таких мереж мало змінилися з 1998 року.

Згорткові нейронні мережі можуть мати багатовимірні шари (в основному

використовуються двомірні, наприклад, в мережах, що обробляють зображення, і тривимірні, наприклад, додаванням декількох колірних каналів для зображення) декількох типів:

- Вхідний шар: вхідне зображення, включаючи кілька колірних каналів.
- Згортковий шар (Convolution): всі нейрони шару, на відміну від персептрона, пов'язані тільки з частиною нейронів попереднього шару.
- Шар Субдискретизації (Pooling, Subsampling): виділення найбільш значущих ознак попереднього шару і значне скорочення розмірності наступних шарів мережі.
- Повнозв'язний шар (Fully-connected): прихований шар штучної нейронної мережі типу персептрон.

Також, після згорткового шару і повнозв'язного шару може застосовуватися функція активації нейрона, яка перетворює сигнал нейрона і формує вихідний сигнал.

Найкращі показники для розпізнавання образів показала саме згорткова нейронна мережа, саме тому її було обрано для побудови власної системи.

Висновки до розділу 3

В результаті аналізу предметної області, а також вимог, які були висунуті при проектуванні, було обрано та обґрунтовано засоби реалізації програмного продукту. Сформовано набір технологій, які будуть використовуватися для створення моделі для нейронної мережі, отримання результатів навчання, виділення та розпізнавання жестів рук людини та управління медіаплеєром за допомогою жестів рук людини.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Система розпізнавання жестів умовно можна поділити на 3 рівні:

Рівень 1: призначений для користувача системи управління - людино-машинний інтерфейс, з якого здійснюється контроль системи.

Рівень 2: інтерфейс, що здійснює навчання нейронної мережі

Рівень 3: інтерфейс, що виводить інформацію про результати навчання

4.1 Опис функціональності системи

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (англ. use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.[25]

Програмний застосунок для програвання музичних творів та розпізнавання жестів людини містить у собі двох акторів – користувача системи та сам аудіоплеєр.

На рисунку 4.1 представлена діаграма прецедентів, яка описує функції та дії акторів у системі.

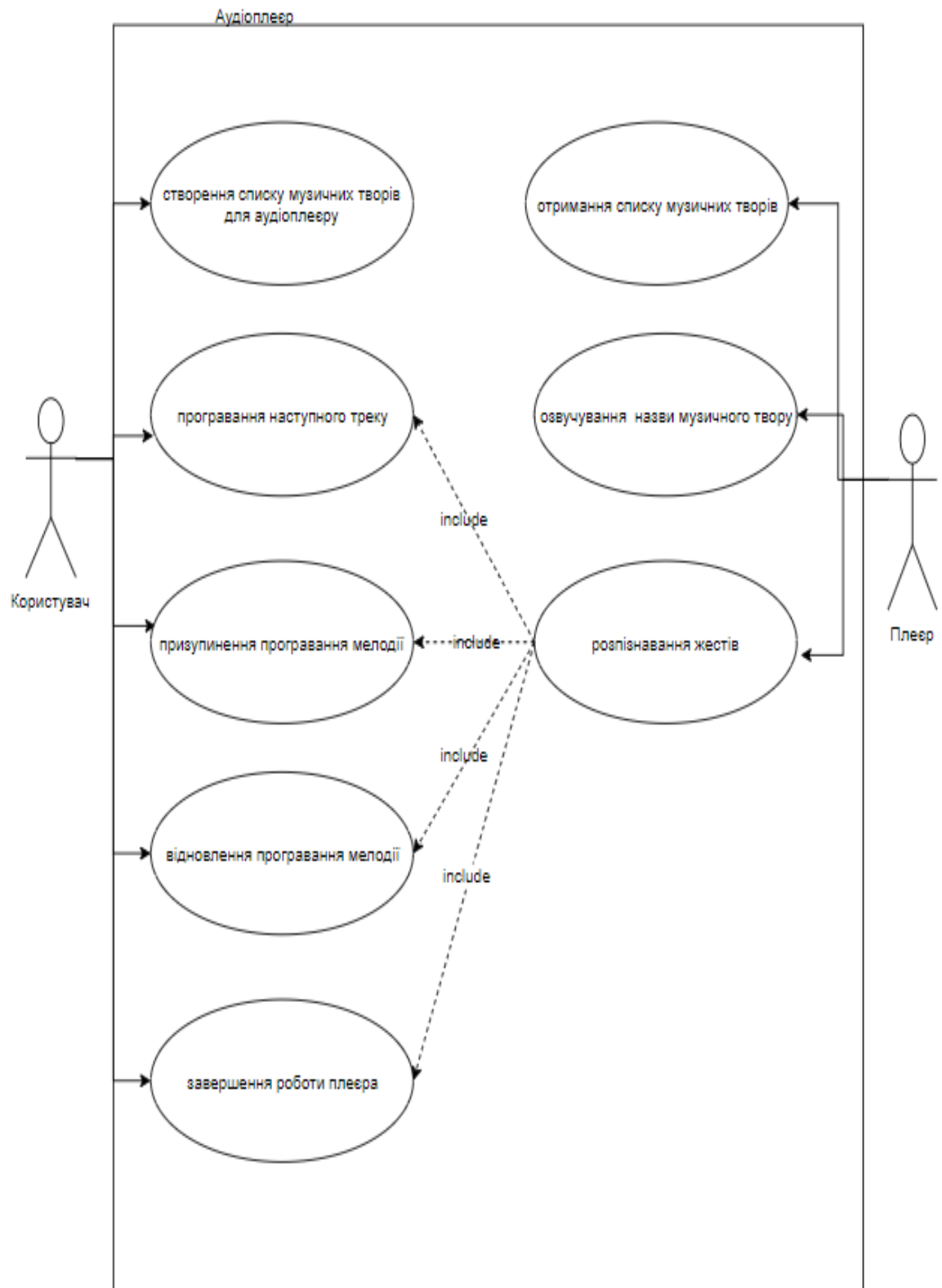


Рисунок 4.1 — Діаграма прецедентів системи

4.2 Розробка методів розпізнавання жестів рук в системах людино-машинної взаємодії

Структурна схема системи розпізнавання жестів з використанням web-камери представлена на рисунку 4.2.

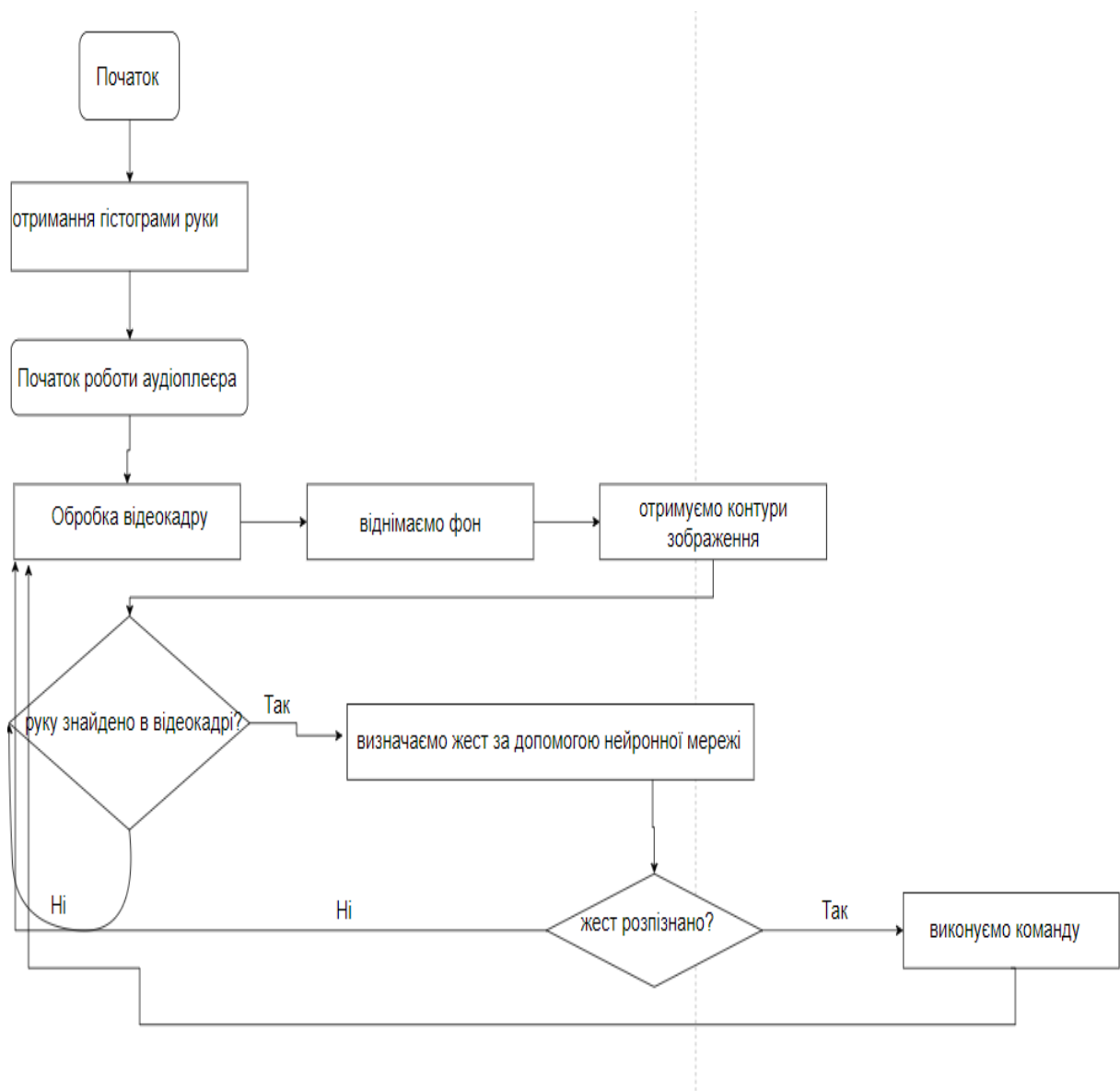


Рисунок 4.2 – Алгоритм роботи додатку

Система складається з web-камери, прикладної програми з нейромережевим модулем розпізнавання і пристроєм виведення інформації. Після зняття зображення данні передаються в модуль прийому відеопотоку, який далі передає опрацьовані

кадри (raw data) в програмний модуль попередньої обробки відеопотоку. Після попередньої обробки відбувається виділення кадрів і їх обробка для пошуку області інтересу (Region of Interest, ROI) на зображенні. Як тільки ROI знайдена, здійснюється пошук ознак гістограми направлених градієнтів (Histogram of Oriented Gradients, HOG) або локальних бінарних шаблонів (Local Binary Patterns, LBP) для подальшої класифікації жесту в модулі. Набір ознак повинен бути достатнім для ідентифікації класу об'єкта, і він представляється у вигляді n -мірного вектора. Сам класифікатор видасть назву жесту якщо жест розпізнано.

4.3 Розробка алгоритму розпізнавання жестів на основі нейромережевої технології

У більшості випадків жести класифікують на два типи: статичні і динамічні. Під статичним жестом розуміється стан долоні в визначений момент часу. Статичні жести іноді називають позами, їх розпізнавання відбувається на статичному зображенні. Динамічні жести є послідовністю статичних жестів, і їх розпізнавання відбувається у відеопослідовності.

Не всі жести не приймають участь у розпізнаванні. У разі якщо у відеопослідовність потрапить жест, який буде не розпізнано – система все одно продовжить роботу, пропускаючи невідомий жест. Алгоритм складається з 5 основних етапів: препроцесингу, генерації ознак, вибір значимих ознак, подача набору ознак на вхід нейронній системі, інтерпретація результатів розпізнавання.

Алгоритм роботи системи розпізнавання жестів представлений на рис. 4.3



Рисунок 4.3 – Алгоритм роботи системи розпізнавання жестів

На етапі попередньої обробки здійснюється підготовка двох навчальних вибірок: позитивної і негативної. Додатково на даному етапі здійснюється переклад кольорового зображення у відтінки сірого, і вирівнювання гістограми, якщо жест розпізнається в відеопослідовності.

Генерація ознак для класифікаторів є важливим етапом, на якому формується вектор ознак. В даному випадку в якості ознак використовується гістограма спрямованих градієнтів, оскільки вона володіє достатньою точністю, а також локальні бінарні шаблони, які виграють у швидкості навчання у каскаді Хаара [28].

Головною концепцією алгоритму гістограм спрямованих градієнтів (histogram

of oriented gradients) є можливість представлення об'єкту на зображенні описом поширення інтенсивності градієнтів або напрямком контурів [18].

Локальні бінарні шаблони (local binary patterns) є описом околиці пікселя на зображенні в двійковому поданні. При цьому задіяний оператор локального бінарного шаблону використовує вісім пікселів, які оточують центральний, приймаючи значення інтенсивності пікселя в центрі в якості деякого «порогового значення» [29].

При відборі значущих ознак існують два підходи: коли немає необхідності у відборі значущих ознак і коли така необхідність є. Перший передбачає, що жест буде знайдений на зображенні як область інтересу і ідентифіційований одним з декількох класифікаторів. В такому випадку доцільно використовувати метод опорних векторів (support vector machine, SVM), оскільки SVM може обробити всі ознаки HOG дескриптора цілком. Або алгоритми посилення композиції класифікаторів (Adam), які справляються з великою кількістю ознак локальних бінарних шаблонів.

У другому випадку з великого обсягу ознак формується компактний вектор, який буде використовуватися в якості вектора ознак зразка.

Відбираються найбільш підходящі з усіх ознаки, причому невідповідні не видаляються, а використовуються при подальших відборах.

В ході експериментів з різними стратегіями формування вектора ознак і відбору значущих ознак обрано такі дескриптори і класифікатори: HOG + SVM і LBP + Adam. Для зниження розмірності вектора ознак вирішено використовувати асоціативну мережу з багатошаровим персептроном (МЛП) з імовірнісним методом головних компонентів.

4.4 Реалізація алгоритму розпізнавання жестів

Для навчання нейронної мережі був обраний наступний набір жестів з датасету Kaggle, що представлені на рисунку 4.4.



Рисунок 4.4 – Жести для розпізнавання

Тобто жест нижньої частини руки, кулак та долоня. Всього в датасеті Kaggle знаходиться більше 6000 фотографій, відібрано по 600 на кожен жест. Зображені жести були отримані під різними кутами нахилу об'єкта в мінливих умовах освітлення, а також з однорідним фоном за допомогою камери LeapMotion.

Вхідною інформацією є відеопотік з web-камери. У відео може бути присутнім частково зашумлений фон і рух на задньому плані, це практично не впливає на працездатність алгоритму.

Різні типи освітлення теж не впливають на ефективність розпізнавання жестів.

Вхідне відео має розширення 640×480 пікселів. Це означає що при обробці відео 10 кадрів в секунду, розпізнавання одного жесту необхідно виконувати за 0,1 секунду.

В результаті за хвилину буде оброблено 600 кадрів, кожен з яких містить $640 \times$

480 = 307200 пікселів. З огляду на те, що запропонований алгоритм аналізує зображення попиксельно, без ефективної апаратної реалізації час роботи алгоритму істотно зростає.

На першому кроці відбувається завантаження каскаду класифікаторів з xml файлу, після цього починається захоплення зображення з web-камери. Відео розділяється на кадри, кожний з яких трансформується в півтонове зображення, додатково проводиться вирівнювання гістограми зображення для нормалізації певних ділянок аднів з різними яркостями.

Після перетворень відбувається пошук руки на зображенні з використанням багатомасштабного детектування, потім здійснюється визначення ознак жесту для подальшого встановлення його типу.

4.5 Створення інсталяційного файлу

Для зручної роботи користувача необхідно створити інсталяційний файл. Для цього в середовищі Anaconda необхідно завантажити бібліотеку pyinstall за допомогою менеджера пакетів pip. Процес загрузки модуля pyinstall показано на малюнку 4.5.

Pyinstall створить файл з розширенням .exe, який можна буде запускати користувачу системи.

Інсталяція (встановлення) — процес встановлення програмного забезпечення на комп'ютер кінцевого користувача. Виконується особливою програмою (пакетним менеджером), присутнім в операційній системі (наприклад, RPM і APT в Linux, Windows Installer в Microsoft Windows), або ж тим, що вже входить до складу самого програмного забезпечення засобом встановлення[28].

```
(base) C:\Users\Natalia>pip install pyinstaller
Collecting pyinstaller
  Downloading https://files.pythonhosted.org/packages/e2/c9/0b44b2ea87ba36395483a672fddd07e6a9cb2b8d3c4a28d7ae76c7e7e1e5
/PyInstaller-3.5.tar.gz (3.5MB)
    | 3.5MB 939kB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing wheel metadata ... done
Collecting pywin32-ctypes>=0.2.0 (from pyinstaller)
  Downloading https://files.pythonhosted.org/packages/9e/4b/3ab2720f1fa4b4bc924ef1932b842edf10007e4547ea8157b0b9fc78599a
/pywin32_ctypes-0.2.0-py2.py3-none-any.whl
Collecting pefile>=2017.8.1 (from pyinstaller)
  Downloading https://files.pythonhosted.org/packages/36/58/acf7f35859d541985f0a6ea3c34baaeafbfaee23642cf11e85fe36453ae77
/pefile-2019.4.18.tar.gz (62kB)
    | 71kB 4.8MB/s
Collecting altgraph (from pyinstaller)
  Downloading https://files.pythonhosted.org/packages/0a/cc/646187eac4b797069e2e6b736f14cdef85dbe405c9bfc7803ef36e4f62ef
/altgraph-0.16.1-py2.py3-none-any.whl
Requirement already satisfied: setuptools in c:\users\natalia\anaconda3\lib\site-packages (from pyinstaller) (40.8.0)
Requirement already satisfied: future in c:\users\natalia\anaconda3\lib\site-packages (from pefile>=2017.8.1->pyinstalle
r) (0.17.1)
Building wheels for collected packages: pyinstaller
  Building wheel for pyinstaller (PEP 517) ... done
  Created wheel for pyinstaller: filename=PyInstaller-3.5-cp37-none-any.whl size=2877932 sha256=b2a5adaf7b86419f34ca3cd3
78580b9f9d4935f157bef78d9aaa54438a1db60c
  Stored in directory: C:\Users\Natalia\AppData\Local\pip\Cache\wheels\c6\4a\0d\9a1c5d3d876eb0675171281c293aed80839115e
2eb022e6d2
Successfully built pyinstaller
Building wheels for collected packages: pefile
  Building wheel for pefile (setup.py) ... done
  Created wheel for pefile: filename=pefile-2019.4.18-cp37-none-any.whl size=60829 sha256=f584e942904887f938f9b60ec8b2ce
871c199e25454387e597ae1e5a6da92e5d
  Stored in directory: C:\Users\Natalia\AppData\Local\pip\Cache\wheels\1c\41\95\4f33011a0c013c872fe6f0f364dc463a25881208
20e40a30d8
Successfully built pefile
Installing collected packages: pywin32-ctypes, pefile, altgraph, pyinstaller
Successfully installed altgraph-0.16.1 pefile-2019.4.18 pyinstaller-3.5 pywin32-ctypes-0.2.0
```

Рисунок 4.5 – Процес встановлення пакету pyinstaller

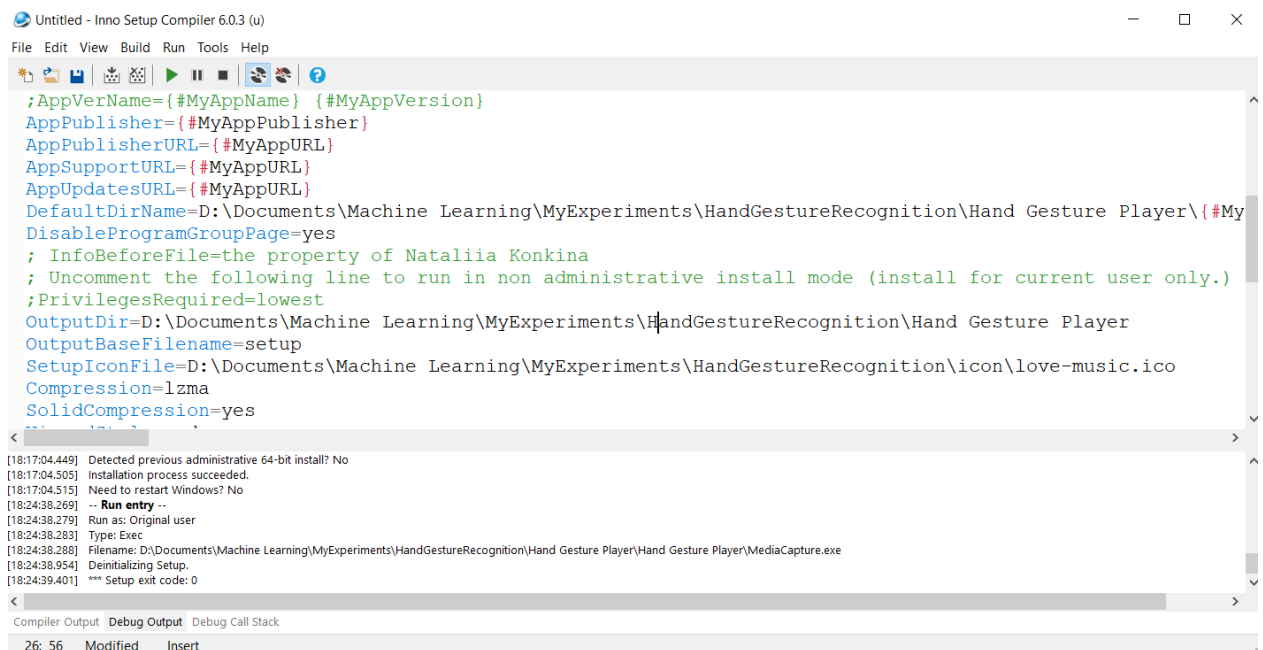
Для створення файлу .exe виконуємо наступну команду, що показана на рисунку 4.6

```
(base) D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition>pyinstaller -F -i "D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition\icon\love-music.ico" popnn_torch.py
```

Рисунок 4.6 – Процес створення .exe файлу

Після цього отримуємо папку з назвою dist, у якій знаходяться файли .exe.

За допомогою програми для упаковки файлів Inno Setup створюємо скрипт як показано на рисунку 4.7. Inno Setup допомагає створити скрипти для інсталяції файлу. У скрипті можна вказати ім'я власника, шлях до директорії, вихідну директорію та інше. Після створення скрипту програма надає можливість створення інсталяційного пакету.



```

;AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
AppPublisherURL={#MyAppURL}
AppSupportURL={#MyAppURL}
AppUpdatesURL={#MyAppURL}
DefaultDirName=D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition\Hand Gesture Player\{#My
DisableProgramGroupPage=yes
; InfoBeforeFile=the property of Nataliia Konkina
; Uncomment the following line to run in non administrative install mode (install for current user only.)
;PrivilegesRequired=lowest
OutputDir=D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition\Hand Gesture Player
OutputBaseFilename=setup
SetupIconFile=D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition\icon\love-music.ico
Compression=lzma
SolidCompression=yes

[18:17:04.449] Detected previous administrative 64-bit install? No
[18:17:04.505] Installation process succeeded.
[18:17:04.515] Need to restart Windows? No
[18:24:38.269] -- Run entry --
[18:24:38.279] Run as: Original user
[18:24:38.283] Type: Exec
[18:24:38.288] Filename: D:\Documents\Machine Learning\MyExperiments\HandGestureRecognition\Hand Gesture Player\Hand Gesture Player\MediaCapture.exe
[18:24:38.954] Deinitializing Setup.
[18:24:39.401] *** Setup exit code: 0
  
```

Рисунок 4.7 – Скрипт для створення файлу установника

Після цього у папці, що прописана у змінній OutputDir знаходимо файл setup.exe. Цей файл допоможе користувачеві встановити додаток та необхідні для нього залежності.

Висновки до розділу 4

У результаті проектування та розробки програмного продукту, який дозволяє розпізнавати жести рук, було надано користувачеві можливість створювати власний список з музичними творами, програвати мелодію, призупиняти/відновлювати програвання або зупиняти програвання мелодії, спостерігати за процесом розпізнавання жестів в режимі реального часу. Було створено клієнтський додаток, який надає графічний інтерфейс для користувача, а також створено власний модуль для навчання нейронної мережі а також модуль для отримання результатів навчання та побудування графіків.

5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Програмний комплекс розроблений для операційних систем Windows та Linux.

5.1 Інсталяція та системні вимоги

Оскільки додаток написано для ОС Windows і Linux, для його коректної роботи додаток необхідно встановити. Для цього запускаємо програму для встановлення setup.exe. Та обираємо мову встановлення. Серед доступних мов – англійська, російська та українська.

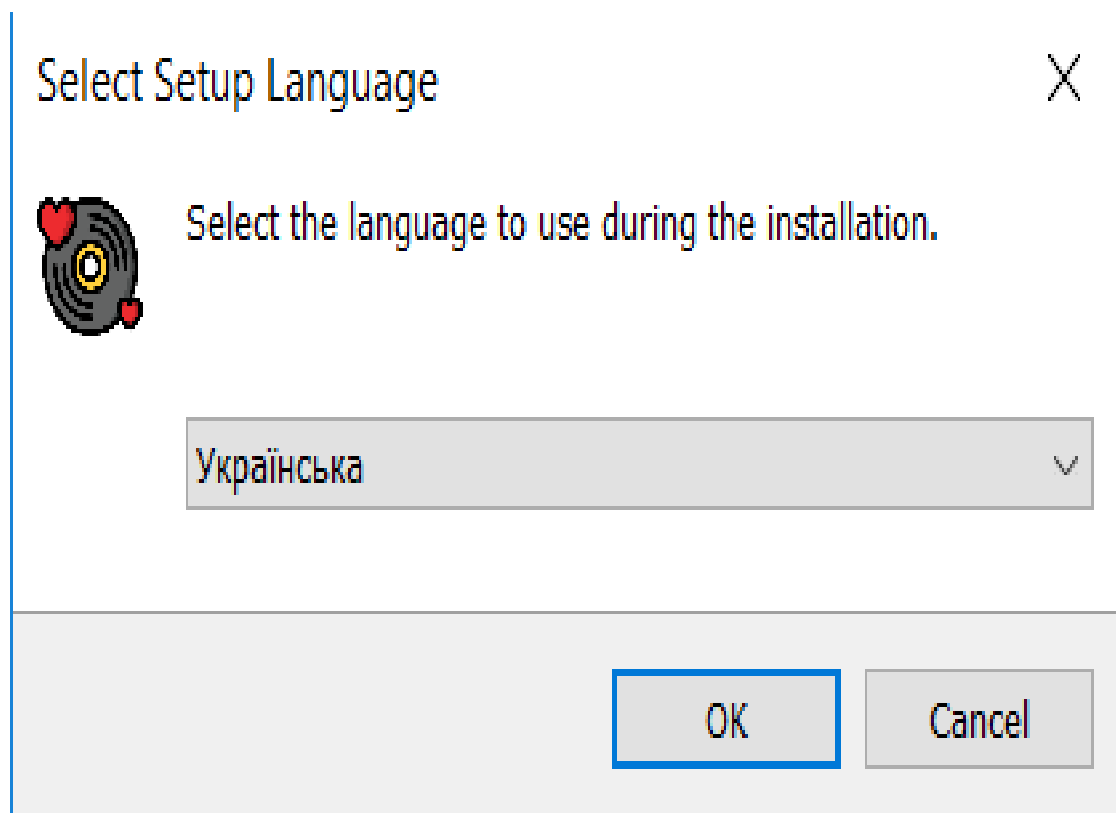


Рисунок 5.1 – Вибір мови встановника

Обираємо папку для установки як показано на рисунку 5.2

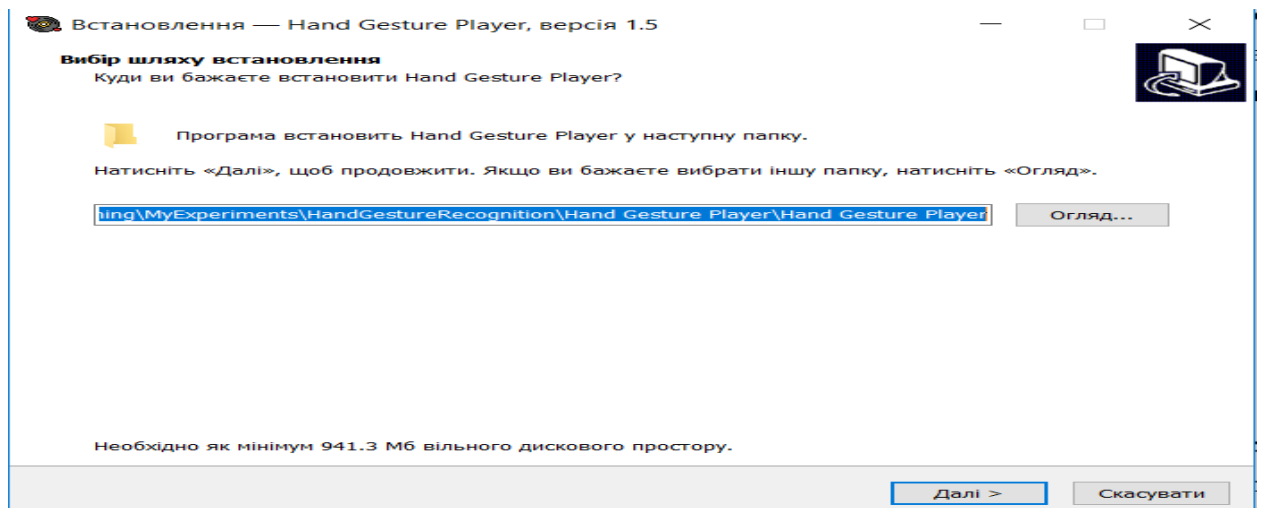


Рисунок 5.2 – Вибір шляху встановлення

Обираємо, чи потрібно створити ярлик на робочому столі(Рисунок 5.1.3)

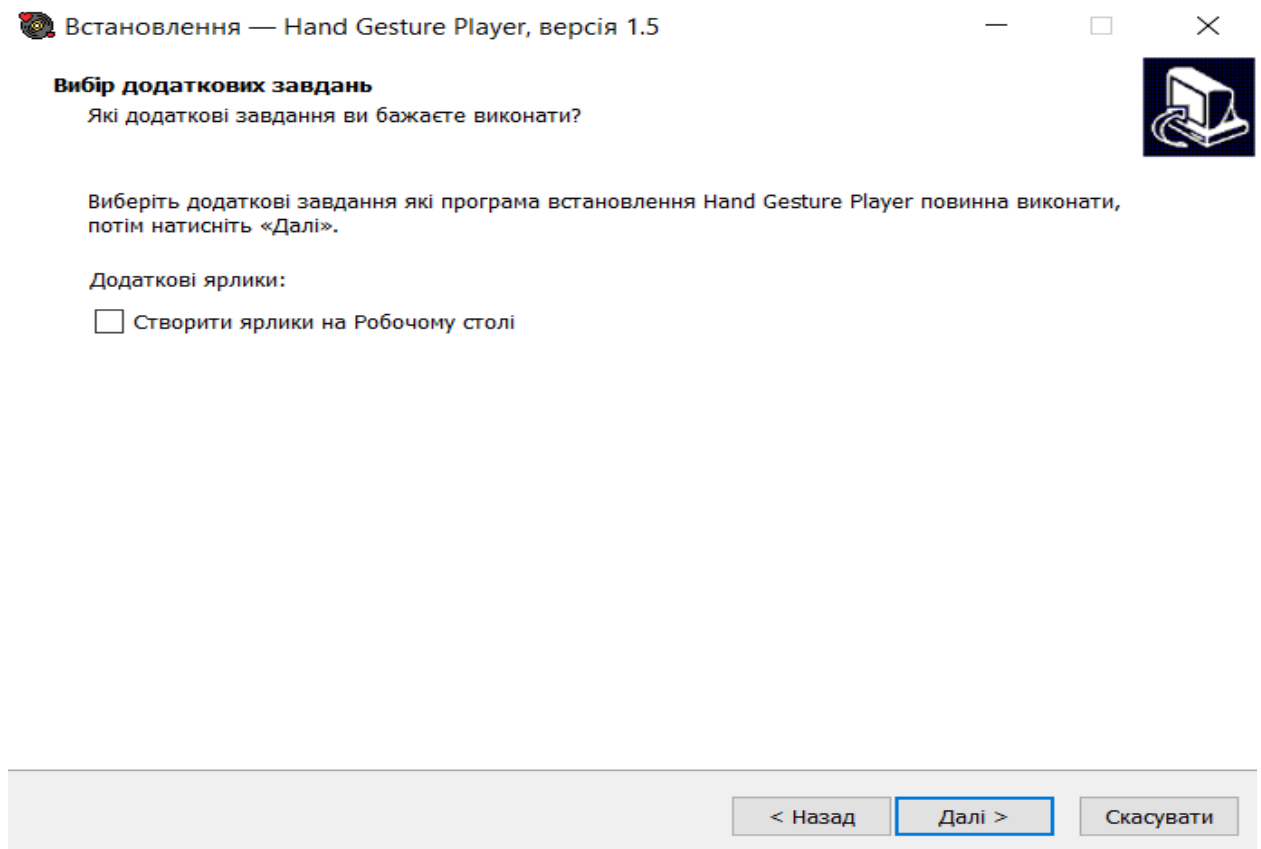


Рисунок 5.3 – Вибір додаткових завдань

Встановлюємо додаток як показано на рисунку 5.4

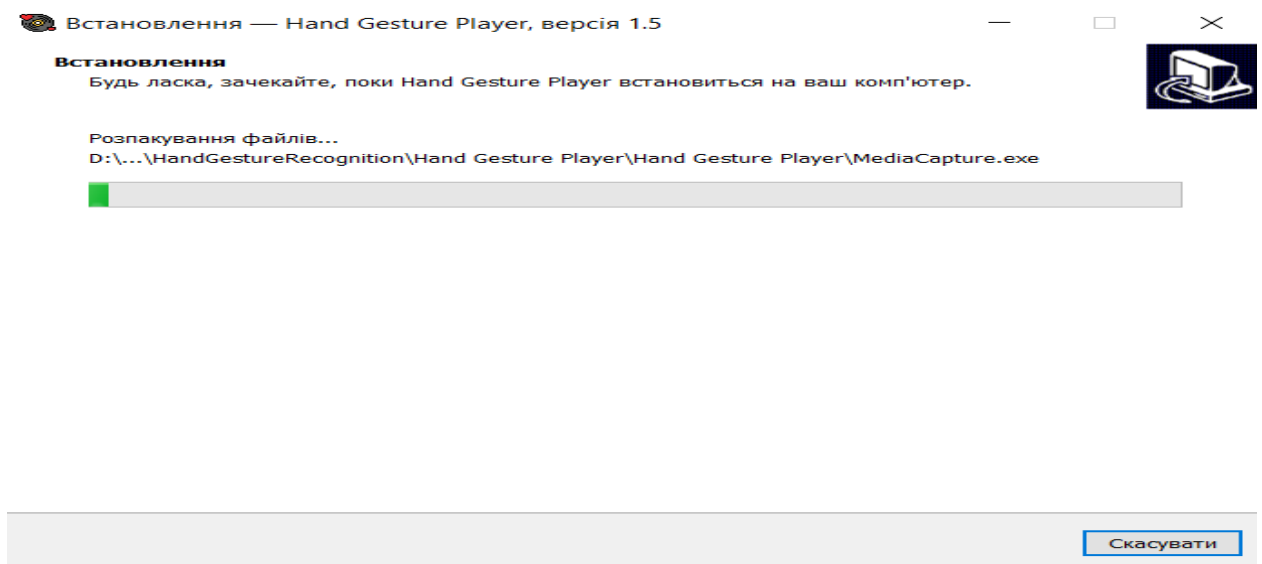


Рисунок 5.4 – Встановлення плеєра

Після завершення встановлення отримуємо про це повідомлення, як показано на рисунку 5.5.

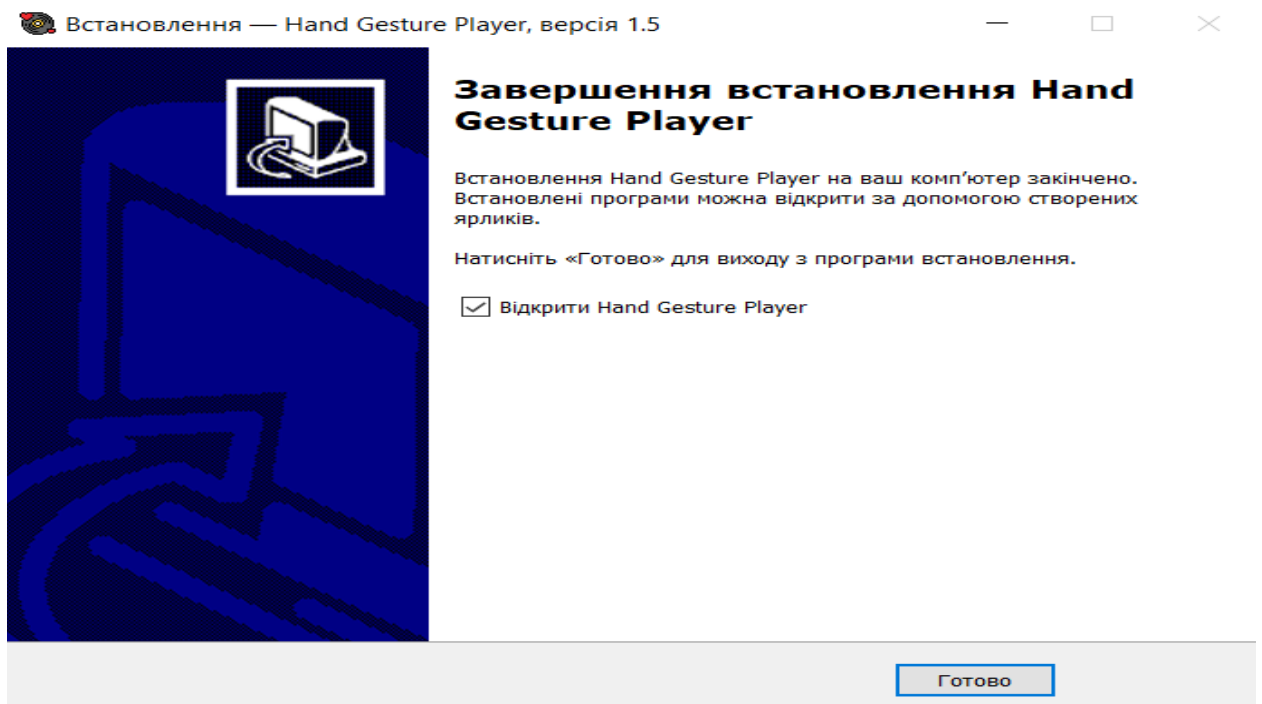


Рисунок 5.5 – Завершення встановлення плеєра

Програма готова для запуску.

Після завершення роботи установника, в папці до встановленого файлу

знайдемо 3 файли:

1. Файл MediaCapture для запуску додатку.
2. Файл даних unins000.dat.
3. Файл для видалення програми.

Рисунок 5.6 показує файли створені встановником.




 MediaCapture	10/20/2019 5:49 PM	Application	961,322 KB
 unins000.dat	10/20/2019 6:17 PM	DAT File	3 KB
 unins000	10/20/2019 6:04 PM	Application	2,516 KB

Рисунок 5.6 – Файли створені встановником

Програма створена для x86 (32-bit) або x64 (64-bit) версії операційної системи Windows або Linux зі встановленим DX 9. Потрібен центральний процесор Intel або AMD від 1.2 GHz, та графічний адаптер з 3D акселератором. Для стабільної роботи додатку необхідно не менше 941.3 Мб оперативної пам'яті і 122 Мб на вінчестері, також необхідна відеокамера. Окрім цього знадобиться монітор з роздільною здатністю SVGA.

5.2 Інструкція з використання програмного продукту

Щоб почати роботи з системою, користувачеві необхідно «сфотографувати» частину руки за допомогою додатку як показано на рисунку 5.7



Рисунок 5.7 – Отримання цвiтової карти

Після цього починає грати музика.

Назва кожного музичного твору озвучується додатком.

Для керування музичним плеєром використовується 3 жесту – долоня, жест для 5 пальців та кулак.

Жести виконують наступні функції:

1. Долоня - перемикає на наступний музичний твір;
2. 5 пальців – перемикає на попередній музичний твір;
3. долоня у руху – призупинення/поновлення мелодії.

Приклад роботи з плеєром наведено на рисунках 5.8, 5.9 та 5.10

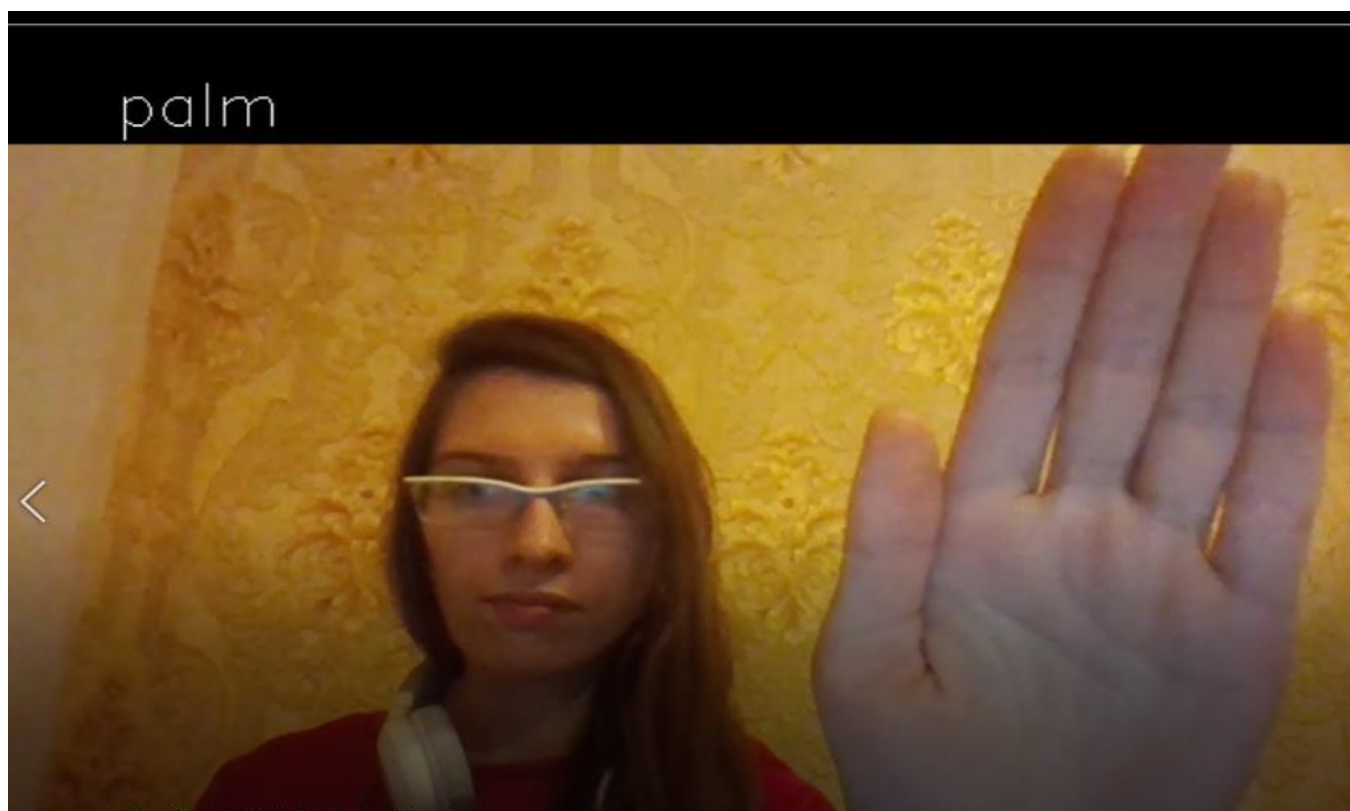


Рисунок 5.8 – Перехід на наступну мелодію за допомогою жесту «Долоня»

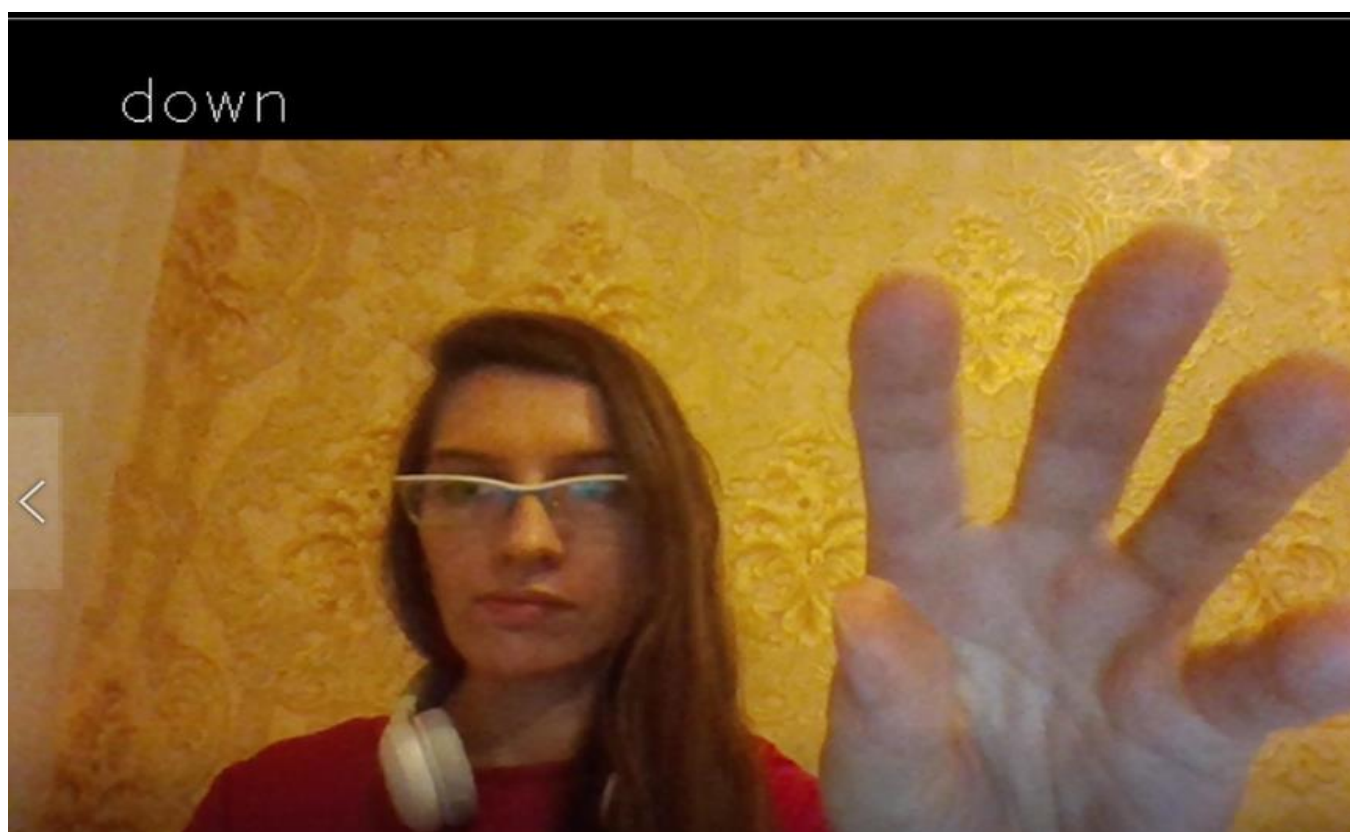


Рисунок 5.9 – Перехід на попередню мелодію за допомогою жесту «5»

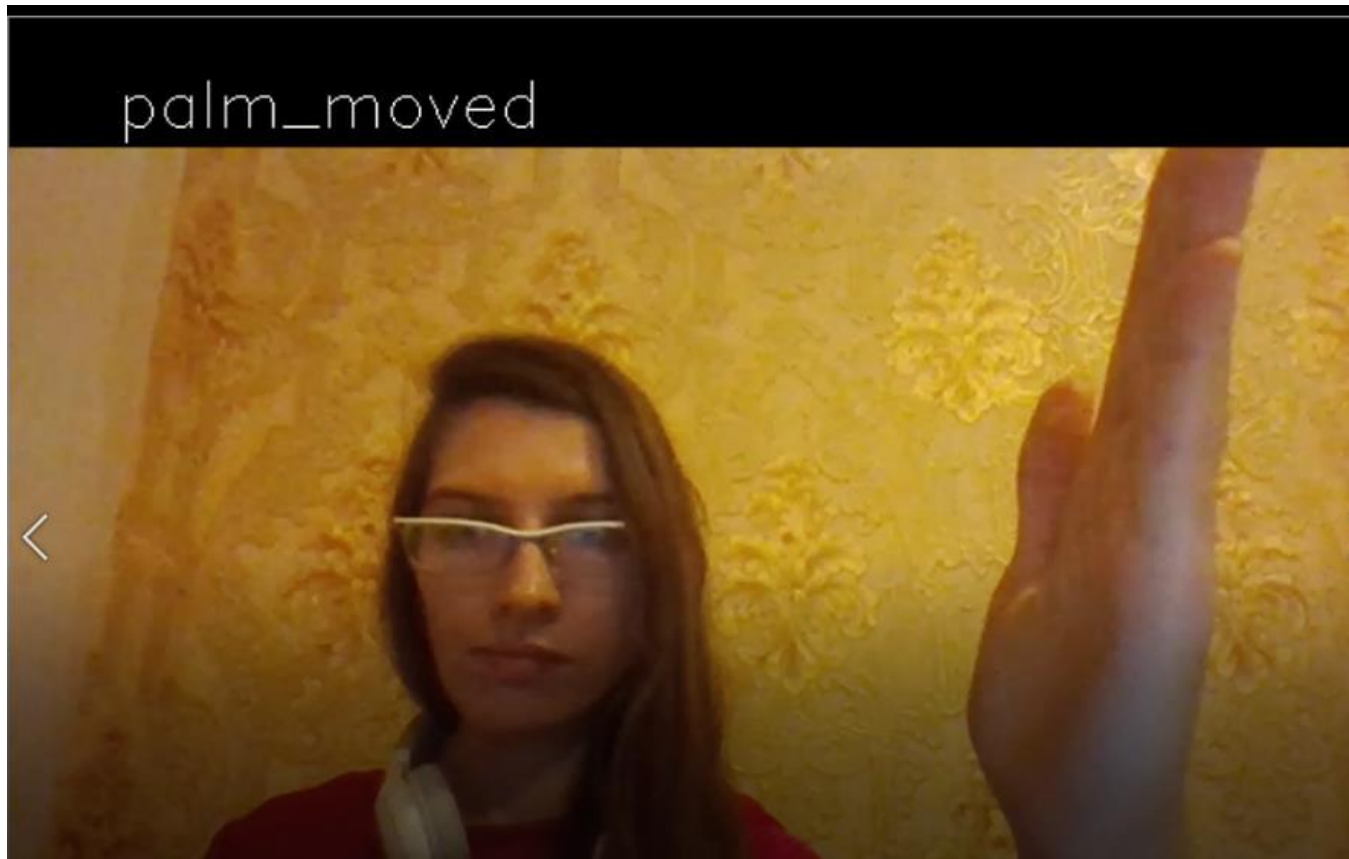


Рисунок 5.10 – Призупинення/поновлення програвання мелодії за допомогою жесту «Долоня у руху»

Висновки до розділу 5

В результаті розробки було створено програмне забезпечення, яке дозволяє розпізнавати жести рук людини для керування музичним плеєром. Основними функціями є зручне управління музичним плеєром – перемикання мелодії, призупинення/поновлення мелодії, зупинення роботи плеєра. Кожна мелодія до програвання озвучується додатком, що надає додатку зручність при роботі з музичним плеєром. Користувач має змогу додавати власні мелодії до плеєра, завдяки чому може прослуховувати різні за уподобанням мелодії.

6. СТАРТАП ПРОЕКТ

Метою розділу є проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження [35]. Проведення маркетингового аналізу передбачає виконання нижченаведених кроків.

6.1 Основні ідеї проекту

В межах підпункту слід проаналізувати та подати у вигляді таблиць:

- зміст запропонованої ідеї;
- можливі напрямки застосування;
- основні переваги, що може отримати користувач товару;
- чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подаються у вигляді таблиці (таблиця 6.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Переваги для користувача
Розпізнавання жестів рук людини	Системи спостереження	Збільшення точності реакції на різноманітні події
	Прямі трансляції, новини	Збирання статистики і реакція на події
	Медичні заклади та заклади догляду за людьми	Розпізнавання та реакція на надзвичайні події

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

— визначення переліку техніко-економічних властивостей та характеристик ідеї [36];

— визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

— проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 6.2).

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик

No п/п		(потенційні) товари/концепції конкурентів		
		Мій проект	Realtime Multi-Person Hand	Paper
1	W слабка сторона	Швидкодія	Відсутність класифікації жестів	Відсутність класифікації жестів
2		Високі вимоги до операційної системи	Високі апаратні вимоги	Відсутність графічного інтерфейсу
3	N нейтральна сторона	Мультиплатформність	Мультиплатформність	Мультиплатформність
4	S сильна сторона	Наявність графічного інтерфейсу	Наявність графічного інтерфейсу	Точність розпізнавання точок рук людини

5		Можливість використання для людей з обмеженими можливостями	Швидкодія	Швидкодія
---	--	---	-----------	-----------

6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту [13]. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/добробити;
- чи доступні такі технології авторам проекту.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	інтерфейс користувача	Python+vlc	Наявна	Доступна безкоштовно
2	Кросплатформені рішення	Python	Наявна	Доступна безкоштовно
3	Розпізнавання жестів рук	Pytorch+openCV	Наявна	Доступна безкоштовно
4	Класифікація наявності людини	Каскади Хаара	Наявна	Доступна безкоштовно
<p>Висновок: проект реалізувати можливо. Обрана технологія реалізації ідеї проекту: інтерфейс користувача, кросплатформений додаток, розпізнавання жестів рук людини, класифікація жестів рук людини</p>				

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів [31].

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

о п/п	Показники стану ринку (найменування)	Характери стика
1	Кількість головних гравців, од	2
2	Загальний обсяг продаж, грн/ум.од	1000000 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Розпізнаван ня жестів людини	Медичні заклади, заклади догляду за людьми, спортивні організатори	Особливості купівлі: компанії заключають довготривалі договори, а стартапери віддають перевагу пробному терміну Використання: компанії вимагають точну роботу програмного продукту, а також широкий набір функцій	Стабільність роботи Невисока ціна Наявність пробного періоду Наявність документації Підтримка необхідних платформ Точність роботи

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 5.6-5.7).

Таблиця 6.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Підходить для нових проектів	Для існуючих проектів виникне потреба переносити базу даних. Продукт більше підходить для нових проектів	Додавання можливості автоматизованого імпорту з різних типів сховищ

2	Обмеженість функцій	Інструмент обмежений наявними функціями і не має деяких функцій, які мають конкуренти	Додавання нових функцій за потреби
---	---------------------	---	------------------------------------

Таблиця 6.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Популярність додатків для ОС	Індустрія постійно набирає обертів	Вихід на глобальний ринок
2	Відсутність повноцінних альтернатив	Існуючі альтернативи не надають повний набір можливостей	Розширення набору функцій

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренції [5].

Пропозиції повинні відповідати на питання “Як просувати продукт”.

Аналіз пропозицій зображено на таблиці 6.8.

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	чиста	Прямі договори з компаніями, презентація продукту на виставках
2. За рівнем конкурентної боротьби - локальний/національний	національний	Публікація статей на міжнародних сайтах

3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	внутрішньогалузева	Розвивати напрямки, нерозвинуті конкурентами
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	товарно-видова	Розповідати про свої переваги перед конкурентом у цій галузі
5. За характером конкурентних переваг - цінова / нецінова	нецінова	Надання функцій, які не надають конкуренти
6. За інтенсивністю - марочна/не марочна	марочна	Надання функцій, які не надають конкуренти

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	- FLUTTER - VLC плеєр	- ML.NET	Мінімізація витрат часу постачальників	Контроль якості	Лояльність споживачів
Висновки:	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	Є можливості виходу на ринок, оскільки існуючі рішення не надають потрібних переваг	Постачальники підлаштовуються під ринок	Клієнти диктують вимоги згідно з умовами експлуатації	Обмеження для роботи на ринку через товари-замінники

На основі аналізу конкуренції, проведеного в п. 3.5 (таблиця 6.9), а також із урахуванням характеристик ідеї проекту (таблиця 6.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиця 6.6-6.7) визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 10.

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Класифікація жестів людини	Існуючі конкуренти не мають можливості класифікувати жести людини.

За визначеними факторами конкурентоспроможності (таблиця 6.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 6.11).

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з даним продуктом						
			-3	-2	-1	0	1	2	3
	Класифікація жестів людини	10							

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (таблиця 6.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: Керування додатком за допомогою жестів Класифікація жестів людини	Слабкі сторони: Повільна швидкість роботи Високі вимоги до операційної системи
Можливості: Відсутність повноцінних альтернатив	Загрози: Сумісність з старими проектами Обмеженість функцій

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Орієнтація поточної моделі на ринок медичних закладів	50 %	40 год
2	Орієнтація поточної моделі на ринок спортивних заходів	20 %	160 год
3	Орієнтація поточної моделі на ринок догляду за людьми	10 %	200 год

Альтернатива, де отримання ресурсів є більш простим та ймовірним – №1 "Орієнтація поточної моделі на ринок медичних закладів", що становить 80 відсотків. Це значення перевищує інші альтернативи.

6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 6.14).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку [25].

Таблиця 6.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Медичні заклади	Готові	Високий	Висока	Просто
2	Закладу догляду за людьми	Потребують недовгих переговорів	Середній	Середня	Складно
3	Спортивні організатори	Потребують переговорів	Низький	Низька	Дуже складно
Які цільові групи обрано: медичні заклади					

Для роботи в обраних сегментах ринку необхідно сформулювати базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

No п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентос- проможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Орієнтація поточної моделі на ринок медичних закладів	Стратегія концентрованого маркетингу	Медичні заклади потребують точності розпізнавання	Стратегія спеціалізації (спирається на диференціацію)
2	Орієнтація поточної моделі на ринок спортивних організаторів	Стратегія концентрованого маркетингу	Спортивні організатори потребують точності розпізнавання та широкий набір вбудованих в програмний продукт функцій	Стратегія спеціалізації (спирається на диференціацію)

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 6.16).

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

No п/п	Чи є проект «першопрохідце м» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	шукати нових споживачів	Так, потрібно збільшити швидкодію програмного продукту та збільшити набір вбудованих в програмний продукт функцій, потрібно зменшити кількість необхідних ресурсів для розміщення серверу, також треба збільшити підтримку мобільних пристроїв для охоплення нового ринку користувачів	Стратегія заняття конкурентної ніші

З обраних сегментів до постачальника (стартап-компанії) та до продукту розробляється стратегія позиціонування (таблиця 6.17). що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
1	Стабільність роботи Невисока ціна Наявність документації Підтримка багатьох платформ	Стратегія спеціалізації (спирається на диференціацію)	Медичні заклади потребують швидкості розробки, яку надає підтримка багатьох платформ даним продуктом	пришвидшення розробки ПЗ підтримка декількох платформ.

6.5 Розроблення маркетингової програми проекту

Для цього у таблиці 6.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Пришвидшення розробки ПЗ	Підтримка декількох платформ	Більшість конкуренти підтримують лише одну платформу.

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 6.19).

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Розпізнавання положення об’єктів на динамічному наборі зображень		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	можливість оптимізації витрат часу	М	Тл
	можливість оптимізації витрат коштів	М	Вр
	відповідність актуальним технологіям	Нм	Тх
	Відповідає вимогам ДСТУ ISO/IEC 25030:2015 Програмна інженерія. Вимоги щодо якості та оцінювання програмного продукту (SQuaRE). Вимоги щодо якості		
	Пакування: готовий до використання інсталятор		
	Марка: Installer Generator		
III. Товар із підкріпленням	Потенційний користувач може ознайомитись з поточним товаром з наукових конференцій та публічних виступів, а також наукових вісників на яких була представлена інформація про даний продукт		
За рахунок чого потенційний товар буде захищено від копіювання: Назва і контент захищені ліцензією MIT; захист інтелектуальної власності			

М/Нм – монотонні або немонотонні;

Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або органолептичні (останній – для продуктів харчування)

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та

третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	27...270 грн	135...270 грн	27000...98000 грн	27...135 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 6.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 6.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнт повинен надаватися в режимах “тріал” та “повний”.	Легість в встановленні, легкість в сплаті послуг	4: Розробник даного продукту - Користувач.	Проводити збут силами посередника.

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 6.22).

Таблиця 6.22. Концепція маркетингових комунікацій

No п/п	Специфіка поведінки цілових клієнтів	Канали комунікацій, якими користуютьс я цілові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Купляють програми через авторизовану мережу	Операційні системи Windows та Linux	Підтримка декількох платформ Пришвидшення розробки ПЗ	Довести, що програмний продукт пришвидшить розробку ПЗ в рази	Розпізнавай , класифікуй, повідомлюй про жести людини

Висновки до розділу 6

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія — це медичні заклади, для яких важлива швидкість розробки та точність розпізнавання.

ВИСНОВКИ

У дисертаційній роботі вирішена задача розпізнавання статичних і динамічних жестів руки, які можуть бути використані для безконтактної взаємодії людини з комп'ютером. Актуальність завдання обумовлена її практичною значущістю, великою кількістю проведених досліджень в цій області, і тим, що якість відомих алгоритмів розпізнавання жестів рук і пальців, як динамічних, так і статичних, з використанням кольорових відеокамер і тривимірних сенсорів все ще залишається недостатньою для побудови практичних систем людино-машинного взаємодії.

В результаті численних експериментів було встановлено, що розроблений алгоритм зменшує кількість помилок під час розпізнавання жестів. Він є універсальним, оскільки може бути доповнений базою зашумлених зображень для збільшення точності розпізнавання на складних зображеннях і в відеопослідовності.

У даній роботі в результаті застосування HOG + SVM кількість помилок склало 2,8% і 2,6% при використанні LBP + Adam. Запропонований алгоритм знизив рівень помилок при розпізнаванні жестів в відеопослідовності web-камери і може використовуватися при створенні природних людино-машинних інтерфейсів, спеціальних систем для глухонімих людей, а також застосовуватися в управлінні програмним забезпеченням на комп'ютері за допомогою жестів.

Для демонстрації роботи алгоритму було створено програмний продукт, який надає такі можливості:

- програвання музичних творів;
- додавання музичних творів для відтворення;
- розпізнавання жестів рук людини;
- керування музичним плеєром за допомогою жестів рук людини;
- забезпечує користувача зручним графічним інтерфейсом, який може

працювати як на комп'ютері, так і на мобільному телефоні.

Проведено огляд методів і засобів розробки програмної системи. Обґрунтовано вибір створення програмної системи. Це дає змогу підвищити гнучкість та зручність системи, як у розробці та супроводі, так і у використанні.

За результатами виконання тестових завдань підтверджена коректність отриманих результатів, отже система відповідає поставленим вимогам.

Результати роботи демонструють, що отриманий програмний продукт можна класифікувати як зручну, безпечну, приватну, та економічну систему на додаток до її великої гнучкості та надійності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Leap Motion, Kinect, Oculus [Электронный ресурс].URL: <https://vc.ru/p/natural-user-interfaces> (дата обращения 16.01.2017). [(2017, Jan. 16). Leap Motion, Kinect, Oculus [Online], (in Russian). Available: <https://vc.ru/p/natural-user-interfaces>]
2. Touch+ make any surface multitouch and more [Элек-тронный ресурс].

URL: <http://www.ractiv.com/> (дата обращения 16.01.2017). [(2017, Jan. 16). Touch+ make any sur-face multitouch and more [Online]. Available: <http://www.ractiv.com/>]

3. Softkinetic, Wikipedia [Электронный ресурс]. URL: <https://en.wikipedia.org/wiki/Softkinetic> (дата обращения 16.01.2017). [(2017, Jan. 16). Softkinetic, Wikipedia [Online]. Available: <https://en.wikipedia.org/wiki/Softkinetic>]

4. Xtion PRO [Электронный ресурс]. URL: https://www.asus.com/3D-Sensor/Xtion_PRO/ (дата обращения 16.01.2017). [(2017, Jan. 16). Xtion PRO [Online]. Available: https://www.asus.com/3D-Sensor/Xtion_PRO/]

5. Как управлять компьютером жестами с помощью веб-камеры [Электронный ресурс]. URL: <http://www.lookatme.ru/mag/how-to/inspiration-howitworks/199949-gesture-control> (дата обращения 16.01.2017). [(2017, Jan. 16). How to control a computer with gestures using a webcam [Online], (in Russian). Available: <http://www.lookatme.ru/mag/how-to/inspiration-howitworks/199949-gesture-control>]

6. Embedded computer vision [Электронный ресурс]. URL: <http://www.eyesight-tech.com/> (дата обращения 16.01.2017). [(2017, Jan. 16). Embedded computer vision [Online]. Available: <http://www.eyesight-tech.com/>]

7. How to control your Mac with hand gestures [Электронный ресурс]. URL: <http://www.macworld.co.uk/how-to/mac-software/control-you-mac-with-hand-gestures-3598714/> (дата обращения 16.01.2017). [(2017, Jan. 16). How to control your Mac with hand gestures [Online]. Available: <http://www.macworld.co.uk/how-to/mac-software/control-you-mac-with-hand-gestures-3598714/>]

8. Алгоритм и методы обнаружения и распознавания жестов руки на видео в режиме реального времени [Электронный ресурс]. URL: <http://sj.kubsau.ru/2014/03/20.pdf> (дата обращения 28.09.2016). [(2016, Sep. 28). Algorithm and detection & identification methods for real-time hand gesture detection into the video [Online], (in Russian). Available: <http://sj.kubsau.ru/2014/03/20.pdf>]

9. Нахождение контуров и операции с ними [Электронный ресурс]. URL: <http://robocraft.ru/blog/computervision/640.html> (дата обращения 28.09.2016). [(2016, Sep. 28). Finding contours and operations with them [Online], (in Russian). Available: <http://robocraft.ru/blog/computervision/640.html>]

10. Vision-based Hand Gesture Recognition for Human-Computer Interaction [Электронный ресурс]. URL: <http://www.ics.forth.gr> (дата обращения 28.09.2016). [(2016, Sep. 28). Vision-based Hand Gesture Recognition for Human-Computer Interaction [Online]. Available: <http://www.ics.forth.gr>]
11. Кузнецов А. И., Сурков А. В., Рудь М. Н. Использование машинного зрения в целях автоматизации и контроля в технических системах [Электронный ресурс]. URL: <http://earchive.tpu.ru/handle/11683/16932> (дата обращения 28.09.2016). [A. I. Kuznetsov, A. V. Surkov, M. N. Rud (2016, Sep. 28). Using computer vision to automate and control in technical systems [Online], (in Russian). Available: <http://earchive.tpu.ru/handle/11683/16932>]
12. Determining optical flow [Электронный ресурс]. URL: http://image.diku.dk/imagecanon/material/HornSchunck_Optical_Flow.pdf (дата обращения 28.09.2016). [(2016, Sep. 28). Determining optical flow [Online]. Available: http://image.diku.dk/imagecanon/material/HornSchunckOptical_Flow.pdf]
13. Шальнов Е., Кононов В., Конушин В. Алгоритм вычитания фона, основанный на поблочных классификаторах // Труды 21-й Международной конференции по компьютерной графике и зрению ГрафиКон'2011. GraphiCon. МАКС Пресс Москва, 2011. С. 227–230 [E. Shalnov, V. Kononov, V. Konushin, “Background subtraction algorithm based on block-based classifiers”, (in Russian), in 21th International Conference on Computer Graphics and Vision Graphi-Con'2011. GraphiCon. MAX Press Moscow, pp. 227-230, 2011.]
14. Background subtraction techniques: a review [Электронный ресурс]. URL: http://profs.sci.univr.it/~cristanm/teaching/sar_files/lezione4/Piccardi.pdf (дата обращения 2.10.2016). [(2016, Oct. 2). Background subtraction techniques: a review [Online]. Available: http://profs.sci.univr.it/~cristanm/teaching/sar_files/lezione4/Piccardi.pdf]
15. Towards Ubiquitous Document Scanning [Электронный ресурс]. URL: https://robau.files.wordpress.com/2010/06/final_report_00012.pdf (дата обращения 2.10.2016). [(2016, Oct. 2). Towards Ubiquitous Document Scanning [Online]. Available: https://robau.files.wordpress.com/2010/06/final_report_00012.pdf]

16. Анализ изображений и видео Лекция 8: Сегментация изображений [Электронный ресурс]. URL: <https://compscicenter.ru/media/slides/> (дата обращения 2.10.2016). [(2016, Oct. 2). Analysis of the images and video Lecture 8: Image segmentation [Online], (in Russian). Available: <https://compscicenter.ru/media/slides/>]
17. Перевод книги Learning OpenCV [Электронный ресурс]. URL: <http://roboforum.ru/forum51/topic9153.html> (дата обращения 2.10.2016). [(2016, Oct. 2). Translation of the book Learning OpenCV [Online], (in Russian). Available: <http://roboforum.ru/forum51/topic9153.html>]
18. Гистограмма направленных градиентов [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Гистограмма_направленных_градиентов (дата обращения 2.10.2016). [(2016, Oct. 2). Histogram of Oriented Gradients [Online], (in Russian)..Available:..https://ru.wikipedia.org/wiki/Гистограмма_направленных_градиентов]
19. Алгоритм машинного обучения AdaBoost [Электронный ресурс]. URL: <http://www.machinelearning.ru> (дата обращения 2.10.2016). [(2016, Oct. 2). AdaBoost machine learning algorithm [Online], (in Russian). Available: <http://www.machinelearning.ru>]
20. Классификация данных методом опорных векторов [Электронный ресурс]. URL: <https://habrahabr.ru/post/105220/> (дата обращения 2.10.2016). [(2016, Oct. 2). Support vector machines data classification [Online], (in Russian). Available: <https://habrahabr.ru/post/105220/>]
21. Уменьшение размерности в данных [Электронный ресурс]. URL: <http://courses.graphicon.ru/files/courses/smisa/2009/lectures/lecture12.pdf> (дата обращения 15.10.2016). [(2016, Oct. 15). Reduction of data dimension [Online], (in Russian). Available: <http://courses.graphicon.ru/files/courses/smisa/2009/lectures/lecture12.pdf>]
22. Искусственная нейронная сеть [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть (дата обращения 15.10.2016). [(2016, Oct. 15). Artificial neural network [Online], (in Russian). Available:..https://ru.wikipedia.org/wiki/Искусственная_нейронная_сеть]

23. Maraqa M., Al-Zboun F., Dhyabat M., Zitar R. A. Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks // Journal of Intelligent Learning Systems and Applications, vol. 4, 2012. [M. Maraqa, F. Al-Zboun, M. Dhyabat, R.A. Zitar, Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks // Journal of Intelli-gent Learning Systems and Applications, vol. 4, 2012.]

24. Murakami K., Taguchi H. Gesture Recognition using Recurrent Neural Networks // ACM Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology (CHI '91), 1999. [K. Murakami, H. Taguchi, Gesture Recognition using Recurrent Neural Networks // ACM Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology (CHI '91), 1999.]

25. Преимущества нейронных сетей [Электронный ресурс]. URL: <http://www.aiportal.ru/articles/neural-networks/advantages.html> (дата обращения 22.10.2016). [(2016, Oct. 22). Advantages of Neural Networks [Online], (in Russian). Available: <http://www.aiportal.ru/articles/neural-networks/advantages.html>]

26. Скрытая марковская модель [Электронный ресурс]. URL: ru.wikipedia.org/wiki/Скрытая_марковская_модель (дата обращения 22.10.2016). [(2016, Oct. 22). Hidden Markov model [Online], (in Russian). Available: ru.wikipedia.org/wiki/Скрытая_марковская_модель]

27. Применение локальных бинарных шаблонов к решению задачи распознавания лиц [Электронный ресурс]. URL: <https://habrahabr.ru/post/193658/> (дата обращения 22.10.2016). [(2016, Oct. 22). The use of local binary patterns to solve problems face recognition [Online], (in Russian). Avail-able: <https://habrahabr.ru/post/193658/>]

28. Cascade classifer [Электронный ресурс]. URL: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html (дата обращения 22.10.2016). [(2016, Oct. 22). Cascade classifer [Online]. Available: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html]

29. Local binary pattern [Электронный ресурс]. URL: <http://docs.opencv.org/2.4/modules/> (дата обращения

22.10.2016). [(2016, Oct. 22). Local binary pattern [Online]. Available: <http://docs.opencv.org/2.4/modules/>]

30. Tutorial: OpenCV haartraining [Электронный ресурс]. URL: <http://note.sonots.com/SciSoftware/haartraining.html> (дата обращения 22.10.2016). [(2016, Oct. 22). Tutorial: OpenCV haartraining [Online]. Available: <http://note.sonots.com/SciSoftware/haartraining.html>]

31. Cascade Classifier Training [Электронный ресурс]. URL: http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html (дата обращения 22.10.2016). [(2016, Oct. 22). Cascade Classifier Training [Online]. Available: http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html]

32. Nianjun Liu, Brian C. Lovell, Gesture Classification Using Hidden Markov Models and Viterbi Path Counting [Электронный ресурс]. URL: <http://staff.itee.uq.edu.au/lovell/aprs/dicta2003/pdf/0273.pdf> (дата обращения 28.10.2016). [L. Nianjun, C. Brian Lovell (2016, Oct. 28). Gesture Classification Using Hidden Markov Models and Viterbi Path Counting [Online]. Available: <http://staff.itee.uq.edu.au/lovell/aprs/dicta2003/pdf/0273.pdf>]

33. Распознавание жестов на видеопоследовательности в режиме реального времени на основе применения метода Виолы – Джонса, алгоритма CAMShift, вейвлет-преобразования и метода главных компонент [Электронный ресурс]. URL: <http://earchive.tpu.ru/handle/11683/4244> (дата обращения 28.10.2016). [(2016, Oct. 28). Real-time hand gesture recognition base on Viola –Jones method, algorithm CAMShift, wavelet transform and principal component analysis [Online], (in Russian). Available: <http://earchive.tpu.ru/handle/11683/4244>]

34. Нгуен Т. Т., Спицын В. Г., Алгоритмическое и программное обеспечение для распознавания формы руки в реальном времени с использованием SURF дескрипторов и нейронной сети [Электронный ресурс]. URL: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2012/v320/i5/09.pdf (дата обращения 28.10.2016). [T. T. Nguen, V. G. Spitsin (2016, Oct. 28). *Algorithmic and the software to recognize the shape of hands in real-time using SURF descriptors and neural network* [Online], (in Russian). Available:

http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2012/v320/i5/09.pdf]

ДОДАТОК

Система розпізнавання жестів для людино-машинної взаємодії

Тези на конференцію “Сучасні проблеми наукового забезпечення енергетики”

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ АПЕПС_ТВ81мпз_18Б

Аркушів 3

Київ 2019

В даний час, час глобалізації та швидкого розвитку методів комунікації між людьми, неможливо залишити осторонь проблему комунікації із зовнішнім світом і соціальної інтеграції людей з обмеженими можливостями. Для вирішення проблем комунікації глухонімих людей можливе використання сучасних технологій з розпізнавання образів.

Метою роботи є розпізнавання жестів людини у реальному часу за допомогою відеокамери та збереження отриманих результатів у базу даних для можливості подальшого аналізу та використання.

Основні завдання розробки системи: провести порівняльний аналітичний огляд існуючих методів захоплення, відстеження і розпізнавання динамічних жестів людини, класифікація жестів що виконуються людиною і вибір алфавіту динамічних жестів, розробка обчислювально ефективної моделі та алгоритм розпізнавання динамічних жестів людини.

У зв'язку з ростом обчислювальних потужностей і появою великих баз зображень стало можливим навчати глибокі нейронні мережі - нейронні мережі з великим числом прихованих шарів. У задачі розпізнавання образів особливого успіху досягли згорткові нейронні мережі (Convolutional Neural Networks), які щороку з 2012 року вигравали змагання ImageNet Large Scale Visual Classification Challenge (ILSVRC).

Альтернативою згортковим мережам для розпізнавання об'єктів на зображенні є метод ознак (каскадів) Хаара. Інакше він ще називається методом Віюлі-Джонса за іменами дослідників, які адаптували метод вейвдетів Хаара до розпізнавання образів.

Поява сенсора Kinect в 2010 році, першої камери глибини, відкрило широкі можливості для створення систем розпізнавання жестів, що надзвичайно підвищило актуальність завдання розпізнавання жестів з використанням камери глибини. А розроблене для Kinect програмне забезпечення виконує визначення положень основних суглобів тіла людини [1].

Для знаходження контуру руки з використанням бібліотек комп'ютерного зору, використовується алгоритм оконтурювання зображення, запропонований Satoshi Suzuki [2], після чого з усіх отриманих контурів вибирається той, в якому знаходиться точка долоні. Алгоритм на цьому кроці, отримуючи на вхід контур руки людини, апроксимує його опуклим багатокутником. Для цього використовується механізм, запропонований Jack Sklansky [3].

Базуючись на проведених дослідженнях, було прийнято рішення про актуальність розробки власного програмно-апаратного комплексу для розпізнавання жестів рук та збереження результатів з метою подальшої обробки та інтерпретації.

Перелік посилань:

1. Zhou Ren, Jingjing Meng, Junsong Yuan, Zhengyou Zhang. Robust Hand Gesture Recognition with Kinect Sensor
// URL: <http://eeeweb.nyu.edu.sg/computervision/Research%20Papers/2011/Robust%20Hand%20Gesture%20Recognition%20with%20Kinect%20Sensor.pdf>
2. Suzuki S., Abe K. Topological Structural Analysis of Digitized Binary Images by Border Following.
3. Jack Sklansky .Finding the convex hull of a simple polygon university of California, Irvine, CA 92717, U.S.A.